

시험에
나오는 것만
공부한다!



두 번 시험보면 한 번은 출제되는 잘못된 문장 찾기 197문제 정보처리기사 필기



1. 소프트웨어 아키텍처와 관련한 설명으로 틀린 것은?

- ① 파이프 필터 아키텍처에서 데이터는 파이프를 통해 양방향으로 흐르며, 필터 이동 시 오버헤드가 발생하지 않는다.
- ② 외부에서 인식할 수 있는 특성이 담긴 소프트웨어의 골격이 되는 기본 구조로 볼 수 있다.
- ③ 데이터 중심 아키텍처는 공유 데이터 저장소를 통해 접근자 간의 통신이 이루어지므로 각 접근자의 수 증가 확장이 용이하다.
- ④ 이해 관계자들의 품질 요구사항을 반영하여 품질 속성을 결정한다.

[해설]

파이프-필터(Pipe-Filter) 패턴은 데이터 통로인 파이프(Pipe)를 이용하여 컴포넌트인 필터(Filter) 간에 데이터를 전송하는 구조로, 단방향 및 양방향 모두 구현할 수 있으며, 필터 간 이동 시 오버헤드가 발생합니다.

2. 애자일 기법에 대한 설명으로 맞지 않은 것은?

- ① 절차와 도구보다 개인과 소통을 중요하게 생각한다.
- ② 계획에 중점을 두어 변경 대응이 난해하다.
- ③ 소프트웨어가 잘 실행되는데 가치를 둔다.
- ④ 고객과의 피드백을 중요하게 생각한다.

[해설]

애자일은 계획을 따르기 보다는 변화에 반응하는 것에 더 가치를 두는 개발 방법론입니다.

3. 디자인 패턴을 이용한 소프트웨어 재사용으로 얻어지는 장점이 아닌 것은?

- ① 소프트웨어 코드의 품질을 향상시킬 수 있다.
- ② 개발 프로세스를 무시할 수 있다.
- ③ 개발자들 사이의 의사소통을 원활하게 할 수 있다.
- ④ 소프트웨어의 품질과 생산성을 향상시킬 수 있다.

[해설]

디자인 패턴을 이용한다고 하더라도 기존의 개발 프로세스를 무시할 수는 없습니다.

4. 요구사항 관리 도구의 필요성으로 틀린 것은?

- ① 요구사항 변경으로 인한 비용 편익 분석
- ② 기존 시스템과 신규 시스템의 성능 비교
- ③ 요구사항 변경의 추적
- ④ 요구사항 변경에 따른 영향 평가

[해설]

요구사항 관리 도구는 사용자의 요구사항 정의 및 변경 사항 등을 관리하는 도구로, 성능 비교를 위해서는 성능 테스트 도구를 사용해야 합니다.

5. 순차 다이어그램(Sequence Diagram)과 관련한 설명으로 틀린 것은?

- ① 객체들의 상호 작용을 나타내기 위해 사용한다.
- ② 시간의 흐름에 따라 객체들이 주고 받는 메시지의 전달 과정을 강조한다.
- ③ 동적 다이어그램보다는 정적 다이어그램에 가깝다.
- ④ 교류 다이어그램(Interaction Diagram)의 한 종류로 볼 수 있다.

[해설]

순차 다이어그램(Sequence Diagram)은 시간의 흐름에 따라 상호 작용하는 개체들을 표현하는 동적 다이어그램입니다.

6. 사용자 인터페이스(UI)의 특징으로 틀린 것은?

- ① 구현하고자 하는 결과의 오류를 최소화한다.
- ② 사용자의 편의성을 높임으로써 작업 시간을 증가시킨다.
- ③ 막연한 작업 기능에 대해 구체적인 방법을 제시하여 준다.
- ④ 사용자 중심의 상호 작용이 되도록 한다.

[해설]

사용자 인터페이스(UI)는 사용자의 편리성과 가독성을 높임으로써 작업 시간을 단축시키고 업무에 대한 이해도를 높여줍니다.

7. 소프트웨어 설계에서 요구사항 분석에 대한 설명으로 틀린 것은?

- ① 소프트웨어가 무엇을 해야 하는가를 추적하여 요구사항 명세를 작성하는 작업이다.
- ② 사용자의 요구를 추출하여 목표를 정하고 어떤 방식으로 해결할 것인지 결정하는 단계이다.
- ③ 소프트웨어 시스템이 사용되는 동안 발견되는 오류를 정리하는 단계이다.
- ④ 소프트웨어 개발의 출발점인 동시에 실질적인 첫 번째 단계이다.

[해설]

소프트웨어 시스템이 사용되는 동안 발견되는 오류를 정리하는 과정은 형상 관리입니다.

8. 설계 기법 중 하향식 설계 방법과 상향식 설계 방법에 대한 비교 설명으로 가장 옳지 않은 것은?

- ① 하향식 설계에서는 통합 검사 시 인터페이스가 이미 정의되어 있어 통합이 간단하다.
- ② 하향식 설계에서 레벨이 낮은 데이터 구조의 세부 사항은 설계 초기 단계에서 필요하다.
- ③ 상향식 설계는 최하위 수준에서 각각의 모듈들을 설계하고 이러한 모듈이 완성되면 이들을 결합하여 검사한다.
- ④ 상향식 설계에서는 인터페이스가 이미 성립되어 있지 않더라도 기능 추가가 쉽다.

[해설]

상향식 설계는 하위 모듈에서 상위 모듈 방향으로 설계하는 것으로 인터페이스가 이미 성립되어 있어 야만 기능 추가가 가능합니다.

9. 디자인 패턴 사용의 장·단점에 대한 설명으로 거리가 먼 것은?

- ① 소프트웨어 구조 파악이 용이하다.
- ② 객체지향 설계 및 구현의 생산성을 높이는데 적합하다.
- ③ 재사용을 위한 개발 시간이 단축된다.
- ④ 절차형 언어와 함께 이용될 때 효율이 극대화된다.

[해설]

디자인 패턴에는 다양한 인터페이스와 객체들을 활용하는 방법들이 제시되어 있기 때문에 절차형 언어보다는 객체지향 언어와 함께 이용될 때 효율이 극대화됩니다.

10. 소프트웨어 아키텍처 모델 중 MVC(Model-View Controller)와 관련한 설명으로 틀린 것은?

- ① MVC 모델은 사용자 인터페이스를 담당하는 계층의 응집도를 높일 수 있고, 여러 개의 다른 UI를 만들어 그 사이에 결합도를 낮출 수 있다.
- ② 모델(Model)은 뷰(View)와 제어(Controller) 사이에서 전달자 역할을 하며, 뷰마다 모델 서브시스템이 각각 하나씩 연결된다.
- ③ 뷰(View)는 모델(Model)에 있는 데이터를 사용자 인터페이스에 보이는 역할을 담당한다.
- ④ 제어(Controller)는 모델(Model)에 명령을 보냄으로써 모델의 상태를 변경할 수 있다.

[해설]

모델(Model)은 서브시스템의 핵심 기능과 데이터를 보관하는 역할을 합니다.

11. 모듈의 독립성을 높이기 위한 결합도(Coupling)와 관련한 설명으로 틀린 것은?

- ① 오류가 발생했을 때 전파되어 다른 오류의 원인이 되는 파문 효과(Ripple Effect)를 최소화해야 한다.
- ② 인터페이스가 정확히 설정되어 있지 않을 경우 불필요한 인터페이스가 나타나 모듈 사이의 의존도는 높아지고 결합도가 증가한다.
- ③ 모듈들이 변수를 공유하여 사용하게 하거나 제어 정보를 교류하게 함으로써 결합도를 낮추어야 한다.
- ④ 다른 모듈과 데이터 교류가 필요한 경우 전역 변수(Global Variable)보다는 매개 변수(Parameter)를 사용하는 것이 결합도를 낮추는 데 도움이 된다.

[해설]

변수를 이용하여 데이터 영역을 공유하는 공통 결합도나, 제어 정보가 이동되는 내용 결합도는 결합의 정도가 강한 결합도입니다.

12. 소프트웨어 공학에서 워크스루(Walkthrough)에 대한 설명으로 틀린 것은?

- ① 사용사례를 확장하여 명세하거나 설계 다이어그램, 원시코드, 테스트 케이스 등에 적용할 수 있다.
- ② 복잡한 알고리즘 또는 반복, 실시간 동작, 병행 처리와 같은 기능이나 동작을 이해하려고 할 때 유용하다.
- ③ 인스펙션(Inspection)과 동일한 의미를 가진다.
- ④ 단순한 테스트 케이스를 이용하여 프로덕트를 수작업으로 수행해 보는 것이다.

[해설]

인스펙션(Inspection)은 워크스루를 발전시킨 형태로, 소프트웨어 개발 단계에서 산출된 결과물의 품질을 평가하고 이를 개선하기 위한 방법 등을 제시합니다.

13. 객체지향 설계에서 정보 은닉(Information Hiding)과 관련한 설명으로 틀린 것은?

- ① 필요하지 않은 정보는 접근할 수 없도록 하여 한 모듈 또는 하부 시스템이 다른 모듈의 구현에 영향을 받지 않게 설계되는 것을 의미한다.
- ② 모듈들 사이의 독립성을 유지시키는 데 도움이 된다.
- ③ 설계에서 은닉되어야 할 기본 정보로는 IP 주소와 같은 물리적 코드, 상세 데이터 구조 등이 있다.
- ④ 모듈 내부의 자료 구조와 접근 동작들에만 수정을 국한하기 때문에 요구사항 등 변화에 따른 수정이 불가능하다.

[해설]

정보 은닉은 모듈이 독립성을 갖게 해주므로, 요구사항 등 변화에 따른 수정이 가능합니다.

14. 소프트웨어 공학에서 워크스루(Walkthrough)에 대한 설명으로 틀린 것은?

- ① 사용사례를 확장하여 명세하거나 설계 다이어그램, 원시코드, 테스트 케이스 등에 적용할 수 있다.
- ② 복잡한 알고리즘 또는 반복, 실시간 동작, 병행 처리와 같은 기능이나 동작을 이해하려고 할 때 유용하다.
- ③ **인스펙션(Inspection)과 동일한 의미를 가진다.**
- ④ 단순한 테스트 케이스를 이용하여 프로덕트를 수작업으로 수행해 보는 것이다.

[해설]

워크스루(Walkthrough)는 ①, ②, ④번과 같은 방법으로 동료 혹은 개발자 그룹 내에서 수행하는 비공식적인 검토 방법인데 반해 인스펙션(Inspection)은 개발자가 제외된 상태에서 전문가가 검토를 수행하는 공식적인 방법으로 워크스루와 인스펙션은 서로 다릅니다.

15. 요구 분석(Requirement Analysis)에 대한 설명으로 틀린 것은?

- ① 요구 분석은 소프트웨어 개발의 실제적인 첫 단계로, 사용자의 요구에 대해 이해하는 단계라 할 수 있다.
- ② 요구 추출(Requirement Elicitation)은 프로젝트 계획 단계에 정의한 문제의 범위 안에 있는 사용자의 요구를 찾는 단계이다.
- ③ 도메인 분석(Domain Analysis)은 요구에 대한 정보를 수집하고 배경을 분석하여 이를 토대로 모델링을 하게 된다.
- ④ **기능적(Functional) 요구에서 시스템 구축에 대한 성능, 보안, 품질, 안정 등에 대한 요구사항을 도출한다.**

[해설]

- 기능적(Functional) 요구사항은 시스템이 무엇을 하는지, 어떤 기능을 하는지 등의 기능이나 수행과 관련된 요구사항입니다.
- 성능, 보안, 품질, 안정 등에 대한 요구사항은 비기능적(Non-functional) 요구사항에 해당합니다.

16. 코드 인스펙션과 관련한 설명으로 틀린 것은?

- ① 프로그램을 수행시켜보는 것 대신에 읽어보고 눈으로 확인하는 방법으로 볼 수 있다.
- ② 코드 품질 향상 기법 중 하나이다.
- ③ **동적 테스트 시에만 활용하는 기법이다.**
- ④ 결함과 함께 코딩 표준 준수 여부, 효율성 등의 다른 품질 이슈를 검사하기도 한다.

[해설]

코드 인스펙션은 정적 테스트 시에만 활용하는 기법입니다.

17. 프로그램 설계도의 하나인 NS Chart에 대한 설명으로 가장 거리가 먼 것은?

- ① 논리의 기술에 중점을 두고 도형을 이용한 표현 방법이다.
- ② 이해하기 쉽고 코드 변환이 용이하다.
- ③ **화살표나 GOTO를 사용하여 이해하기 쉽다.**
- ④ 연속, 선택, 반복 등의 제어 논리 구조를 표현한다.

[해설]

N-S 차트는 GOTO나 화살표를 사용하지 않습니다.

18. 사용자 인터페이스(User Interface)에 대한 설명으로 틀린 것은?

- ① 사용자와 시스템이 정보를 주고받는 상호 작용이 잘 이루어지도록 하는 장치나 소프트웨어를 의미한다.
- ② 편리한 유지보수를 위해 개발자 중심으로 설계되어야 한다.
- ③ 배우기가 용이하고 쉽게 사용할 수 있도록 만들어져야 한다.
- ④ 사용자 요구사항이 UI에 반영될 수 있도록 구성해야 한다.

[해설]

사용자 인터페이스(UI)는 사용자가 쉽게 이해하고 편리하게 사용할 수 있도록 사용자 중심으로 설계되어야 합니다.

19. 소프트웨어 개발에서 모듈(Module)이 되기 위한 주요 특징에 해당하지 않는 것은?

- ① 다른 것들과 구별될 수 있는 독립적인 기능을 가진 단위(Unit)이다.
- ② 독립적인 컴파일이 가능하다.
- ③ 유일한 이름을 가져야 한다.
- ④ 다른 모듈에서의 접근이 불가능해야 한다.

[해설]

각 모듈들은 상호 작용을 통해 더 큰 시스템을 구성해야 하므로 모듈은 상호 접근이 가능해야 합니다.

20. 공학적으로 잘된 소프트웨어(Well Engineered Software)의 설명 중 틀린 것은?

- ① 소프트웨어는 유지보수가 용이해야 한다.
- ② 소프트웨어는 신뢰성이 높아야 한다.
- ③ 소프트웨어는 사용자 수준에 무관하게 일관된 인터페이스를 제공해야 한다.
- ④ 소프트웨어는 충분한 테스트를 거쳐야 한다.

[해설]

공학적으로 잘 작성된 소프트웨어는 사용자의 요구사항을 고려하여, 수준에 맞는 쉬운 인터페이스를 제공합니다.

21. 소프트웨어 모델링과 관련한 설명으로 틀린 것은?

- ① 모델링 작업의 결과물은 다른 모델링 작업에 영향을 줄 수 없다.
- ② 구조적 방법론에서는 DFD(Data Flow Diagram), DD(Data Dictionary) 등을 사용하여 요구사항의 결과를 표현한다.
- ③ 객체지향 방법론에서는 UML 표기법을 사용한다.
- ④ 소프트웨어 모델을 사용할 경우 개발될 소프트웨어에 대한 이해도 및 이해 당사자 간의 의사소통 향상에 도움이 된다.

[해설]

모델링 작업의 결과물은 다른 모델링 작업에 영향을 줄 수 있습니다.

22. 요구사항 명세 기법에 대한 설명으로 틀린 것은?

- ① 비정형 명세 기법은 사용자의 요구를 표현할 때 자연어를 기반으로 서술한다.
- ② 비정형 명세 기법은 사용자의 요구를 표현할 때 Z 비정형 명세 기법을 사용한다.
- ③ 정형 명세 기법은 사용자의 요구를 표현할 때 수학적 원리와 표기법을 이용한다.
- ④ 정형 명세 기법은 비정형 명세 기법에 비해 표현이 간결하다.

[해설]

Z 명세 기법은 비정형 명세가 아니라 정형 명세 기법의 한 종류입니다.

23. 분산 시스템에서의 미들웨어(Middleware)와 관련한 설명으로 틀린 것은?

- ① 분산 시스템에서 다양한 부분을 관리하고 통신하며 데이터를 교환하게 해주는 소프트웨어로 볼 수 있다.
- ② 위치 투명성(Location Transparency)을 제공한다.
- ③ 분산 시스템의 여러 컴포넌트가 요구하는 재사용 가능한 서비스의 구현을 제공한다.
- ④ 애플리케이션과 사용자 사이에서만 분산 서비스를 제공한다.

[해설]

애플리케이션과 사용자 사이뿐만 아니라 클라이언트와 서버, 운영체제와 응용 프로그램과 같이 두 시스템 사이에서 다양한 서비스를 제공하는 소프트웨어를 미들웨어(Middleware)라고 합니다.

24. GoF(Gang of Four) 디자인 패턴과 관련한 설명으로 틀린 것은?

- ① 디자인 패턴을 목적(Purpose)으로 분류할 때 생성, 구조, 행위로 분류할 수 있다.
- ② Strategy 패턴은 대표적인 구조 패턴으로 인스턴스를 복제하여 사용하는 구조를 말한다.
- ③ 행위 패턴은 클래스나 객체들이 상호작용하는 방법과 책임을 분산하는 방법을 정의한다.
- ④ Singleton 패턴은 특정 클래스의 인스턴스가 오직 하나임을 보장하고, 이 인스턴스에 대한 접근 방법을 제공한다.

[해설]

- 전략(Strategy) 패턴은 동일한 계열의 알고리즘들을 개별적으로 캡슐화하여 상호 교환할 수 있게 정의하는 행위 패턴입니다.
- 인스턴스를 복제하여 사용하는 패턴은 생성 패턴의 프로토타입(Prototype) 패턴입니다.

25. 객체지향 분석 기법과 관련한 설명으로 틀린 것은?

- ① 동적 모델링 기법이 사용될 수 있다.
- ② 기능 중심으로 시스템을 파악하며 순차적인 처리가 중요시되는 하향식(Top-down) 방식으로 볼 수 있다.
- ③ 데이터와 행위를 하나로 묶어 객체를 정의하고 추상화시키는 작업이라 할 수 있다.
- ④ 코드 재사용에 의한 프로그램 생산성 향상 및 요구에 따른 시스템의 쉬운 변경이 가능하다.

[해설]

객체지향 분석 기법은 순차적인 처리가 아닌 부품을 조립하듯 클래스를 조립하는 방식으로 처리하며, 하향식 및 상향식 방식 모두 사용할 수 있습니다.

26. 소프트웨어 모듈화의 장점이 아닌 것은?

- ① 오류의 파급 효과를 최소화한다.
- ② 기능의 분리가 가능하여 인터페이스가 복잡하다.
- ③ 모듈의 재사용 가능으로 개발과 유지보수가 용이하다.
- ④ 프로그램의 효율적인 관리가 가능하다.

[해설]

모듈화의 장점은 기능의 분리가 가능하여 인터페이스가 단순해지는 것입니다.

27. 객체에 대한 설명으로 틀린 것은?

- ① 객체는 상태, 동작, 고유 식별자를 가진 모든 것이라 할 수 있다.
- ② 객체는 공통 속성을 공유하는 클래스들의 집합이다.
- ③ 객체는 필요한 자료 구조와 이에 수행되는 함수들을 가진 하나의 독립된 존재이다.
- ④ 객체의 상태는 속성값에 의해 정의된다.

[해설]

객체는 클래스의 집합이 아니라 클래스가 공통된 속성과 연산(행위)을 갖는 객체의 집합입니다.

28. CASE(Computer-Aided Software Engineering) 도구에 대한 설명으로 거리가 먼 것은?

- ① 소프트웨어 개발 과정의 일부 또는 전체를 자동화하기 위한 도구이다.
- ② 표준화된 개발 환경 구축 및 문서 자동화 기능을 제공한다.
- ③ 작업 과정 및 데이터 공유를 통해 작업자 간 커뮤니케이션을 증대한다.
- ④ 2000년대 이후 소개되었으며, 객체지향 시스템에 한해 효과적으로 활용된다.

[해설]

CASE는 객체지향 시스템뿐만 아니라 구조적 시스템 등 다양한 시스템에서 활용되는 자동화 도구입니다.

29. 애자일 소프트웨어 개발 기법의 가치가 아닌 것은?

- ① 프로세스의 도구보다는 개인과 상호 작용에 더 가치를 둔다.
- ② 계약 협상보다는 고객과의 협업에 더 가치를 둔다.
- ③ 실제 작동하는 소프트웨어보다는 이해하기 좋은 문서에 더 가치를 둔다.
- ④ 계획을 따르기보다는 변화에 대응하는 것에 더 가치를 둔다.

[해설]

애자일(Agile)은 방대한 문서보다는 실행되는 소프트웨어에 더 가치를 둡니다.

30. 분산 시스템을 위한 마스터-슬레이브(Master-Slave) 아키텍처에 대한 설명으로 틀린 것은?

- ① 일반적으로 실시간 시스템에서 사용된다.
- ② 마스터 프로세스는 일반적으로 연산, 통신, 조정을 책임진다.
- ③ 슬레이브 프로세스는 데이터 수집 기능을 수행할 수 없다.
- ④ 마스터 프로세스는 슬레이브 프로세스들을 제어할 수 있다.

[해설]

슬레이브 프로세스에서는 마스터 프로세스에서 수행하는 연산, 통신, 제어 등의 기능을 제외하고는 별도로 제한되는 기능은 없습니다.

31. 소프트웨어 개발에 이용되는 모델(Model)에 대한 설명 중 거리가 먼 것은?

- ① 모델은 개발 대상을 추상화하고 기호나 그림 등으로 시각적으로 표현한다.
- ② 모델을 통해 소프트웨어에 대한 이해도를 향상시킬 수 있다.
- ③ 모델을 통해 이해 당사자 간의 의사소통이 향상된다.
- ④ 모델을 통해 향후 개발될 시스템의 유추는 불가능하다.

[해설]

소프트웨어 개발에 이용되는 모델(Model)은 사용자의 요구사항을 정확히 파악하기 위해 실제 개발될 소프트웨어에 대한 견본 형태로 만든 것이므로 최종 결과물에 대한 예측이 가능합니다.

32. 객체지향 개념에서 다형성(Polymorphism)과 관련한 설명으로 틀린 것은?

- ① 다형성은 현재 코드를 변경하지 않고 새로운 클래스를 쉽게 추가할 수 있게 한다.
- ② 다형성이란 여러 가지 형태를 가지고 있다는 의미로, 여러 형태를 받아들일 수 있는 특징을 말한다.
- ③ 메소드 오버라이딩(Overriding)은 상위 클래스에서 정의한 일반 메소드의 구현을 하위 클래스에서 무시하고 재정의할 수 있다.
- ④ 메소드 오버로딩(Overloading)의 경우 매개 변수 타입은 동일하지만 메소드명을 다르게 함으로써 구현, 구분할 수 있다.

[해설]

메소드 오버로딩(Overloading)은 메소드명은 같지만 매개 변수의 개수나 타입을 다르게 함으로써 구현, 구분할 수 있습니다.

33. 정형 기술 검토(FTR)의 지침으로 틀린 것은?

- ① 의제를 제한한다.
- ② 논쟁과 반박을 제한한다.
- ③ 문제 영역을 명확히 표현한다.
- ④ 참가자의 수를 제한하지 않는다.

[해설]

정형 기술 검토는 의제와 참가자의 수를 제한합니다.

34. 사용자 인터페이스를 설계할 경우 고려해야 할 가이드라인과 가장 거리가 먼 것은?

- ① 심미성을 사용성보다 우선하여 설계해야 한다.
- ② 효율성을 높이게 설계해야 한다.
- ③ 발생하는 오류를 쉽게 수정할 수 있어야 한다.
- ④ 사용자에게 피드백을 제공해야 한다.

[해설]

사용자 인터페이스(UI)를 설계할 경우 심미성보다는 사용성을 우선하여 설계해야 합니다.

35. UML(Unified Modeling Language)에 대한 설명 중 틀린 것은?

- ① 기능적 모델은 사용자 측면에서 본 시스템 기능이며, UML에서는 Use case Diagram을 사용한다.
- ② 정적 모델은 객체, 속성, 연관 관계, 오퍼레이션의 시스템의 구조를 나타내며, UML에서는 Class Diagram을 사용한다.
- ③ 동적 모델은 시스템의 내부 동작을 말하며, UML에서는 Sequence Diagram, State Diagram, Activity Diagram을 사용한다.
- ④ State Diagram은 객체들 사이의 메시지 교환을 나타내며, Sequence Diagram은 하나의 객체가 가진 상태와 그 상태의 변화에 의한 동작 순서를 나타낸다.

[해설]

- 상태 다이어그램(State Diagram)은 하나의 객체가 가진 상태와 그 상태의 변화에 의한 동작 순서를 나타냅니다.
- 시퀀스 다이어그램(Sequence Diagram)은 객체들 사이의 메시지 교환을 나타냅니다.

36. 소프트웨어 품질 관련 국제 표준인 ISO/IEC 25000에 관한 설명으로 옳지 않은 것은?

- ① 소프트웨어 품질 평가를 위한 소프트웨어 품질 평가 통합 모델 표준이다.
- ② System and Software Quality Requirements and Evaluation으로 줄여서 SQuaRE라고도 한다.
- ③ ISO/IEC 2501n에서는 소프트웨어의 내부 측정, 외부 측정, 사용 품질 측정, 품질 측정 요소 등을 다룬다.
- ④ 기존 소프트웨어 품질 평가 모델과 소프트웨어 평가 절차 모델인 ISO/IEC 9126과 ISO/IEC 14598을 통합하였다.

[해설]

- ISO/IEC 2501n에서는 소프트웨어의 내부 및 외부 품질과 사용 품질에 대한 모델 등 품질 모델 부분을 다룹니다.
- 소프트웨어의 내부 측정, 외부 측정, 사용 품질 측정, 품질 측정 요소 등 품질 측정 부분을 다루는 것은 ISO/IEC 2502n입니다.

37. UI의 설계 지침으로 틀린 것은?

- ① 이해하기 편하고 쉽게 사용할 수 있는 환경을 제공해야 한다.
- ② 주요 기능을 메인 화면에 노출하여 조작이 쉽도록 하여야 한다.
- ③ 치명적인 오류에 대한 부정적인 사항은 사용자가 인지할 수 없도록 한다.
- ④ 사용자의 직무, 연령, 성별 등 다양한 계층을 수용하여야 한다.

[해설]

오류가 발생하면 사용자가 쉽게 인지할 수 있도록 설계해야 합니다.

38. 소프트웨어 개발 방법 중 요구사항 분석(Requirements Analysis)과 거리가 먼 것은?

- ① 비용과 일정에 대한 제약 설정
- ② 타당성 조사
- ③ 요구사항 정의 문서화
- ④ 설계 명세서 작성

[해설]

설계 명세서를 작성하는 과정은 요구사항 명세(Requirement Specification)에 해당합니다.

39. 요구사항 검증(Requirements Validation)과 관련한 설명으로 틀린 것은?

- ① 요구사항이 고객이 정말 원하는 시스템을 제대로 정의하고 있는지 점검하는 과정이다.
- ② 개발 완료 이후에 문제점이 발견될 경우 막대한 재작업 비용이 들 수 있기 때문에 요구사항 검증은 매우 중요하다.
- ③ 요구사항이 실제 요구를 반영하는지, 문서상의 요구사항은 서로 상충되지 않는지 등을 점검한다.
- ④ 요구사항 검증 과정을 통해 모든 요구사항 문제를 발견할 수 있다.

[해설]

요구사항 검증 과정을 정밀하게 수행하면 대부분의 문제를 발견할 수는 있으나 모든 문제를 발견할 수 있다고 말하기에는 어렵습니다.

40. 유스케이스 다이어그램(Use Case Diagram)에 관련된 내용으로 틀린 것은?

- ① 시스템과 상호 작용하는 외부 시스템은 액터로 파악해서는 안된다.
- ② 유스케이스는 사용자 측면에서의 요구사항으로, 사용자가 원하는 목표를 달성하기 위해 수행할 내용을 기술한다.
- ③ 시스템 액터는 다른 프로젝트에서 이미 개발되어 사용되고 있으며, 본 시스템과 데이터를 주고받는 등 서로 연동되는 시스템을 말한다.
- ④ 액터가 인식할 수 없는 시스템 내부의 기능을 하나의 유스케이스로 파악해서는 안된다.

[해설]

시스템과 상호 작용하는 모든 외부 요소를 액터라고 합니다.

41. 객체지향 기법의 캡슐화(Encapsulation)에 대한 설명으로 틀린 것은?

- ① 인터페이스가 단순화 된다.
- ② 소프트웨어 재사용성이 높아진다.
- ③ 변경 발생 시 오류의 파급 효과가 적다.
- ④ 상위 클래스의 모든 속성과 연산을 하위 클래스가 물려받는 것을 의미한다.

[해설]

④번은 상속(Inheritance)에 대한 설명입니다.

42. User Interface 설계 시 오류 메시지나 경고에 관한 지침으로 가장 거리가 먼 것은?

- ① 메시지는 이해하기 쉬워야 한다.
- ② 오류로부터 회복을 위한 구체적인 설명이 제공되어야 한다.
- ③ 오류로 인해 발생할 수 있는 부정적인 내용을 적극적으로 사용자들에게 알려야 한다.
- ④ 소리나 색의 사용을 줄이고 텍스트로만 전달하도록 한다.

[해설]

오류 메시지나 경고는 소리나 색 등을 이용하여 듣거나 보기 쉽게 의미를 전달해야 합니다.

43. 익스트림 프로그래밍(XP)에 대한 설명으로 틀린 것은?

- ① 빠른 개발을 위해 테스트를 수행하지 않는다.
- ② 사용자의 요구사항은 언제든지 변할 수 있다.
- ③ 고객과 직접 대면하며 요구사항을 이야기하기 위해 사용자 스토리(User Story)를 활용할 수 있다.
- ④ 기존의 방법론에 비해 실용성(Pragmatism)을 강조한 것이라고 볼 수 있다.

[해설]

익스트림 프로그래밍(XP; eXtreme Programming)은 고객의 요구 변화에 신속하게 대응하기 위해 릴리즈 기간을 가능한 짧게 반복하는데, 릴리즈 기간 동안 지속적으로 테스트가 진행될 수 있도록 자동화된 테스트 구조를 사용합니다.

44. 바람직한 소프트웨어 설계 지침이 아닌 것은?

- ① 모듈의 기능을 예측할 수 있도록 정의한다.
- ② 이식성을 고려한다.
- ③ 적당한 모듈의 크기를 유지한다.
- ④ 가능한 모듈을 독립적으로 생성하고 결합도를 최대화한다.

[해설]

바람직한 소프트웨어를 설계하기 위해서는 모듈의 결합도는 줄이고 응집도는 높여야 합니다.

45. DFD(Data Flow Diagram)에 대한 설명으로 틀린 것은?

- ① 자료 흐름 그래프 또는 버블(Bubble) 차트라고도 한다.
- ② 구조적 분석 기법에 이용된다.
- ③ 시간 흐름을 명확하게 표현할 수 있다.
- ④ DFD의 요소는 화살표, 원, 사각형, 직선(단선/이중선)으로 표시한다.

[해설]

자료 흐름도(DFD)는 자료의 흐름 및 변환 과정과 기능을 도형 중심으로 기술하는 방법으로, 시간의 흐름은 명확히 표현되지 않습니다.

46. CASE(Computer Aided Software Engineering)에 대한 설명으로 틀린 것은?

- ① 소프트웨어 모듈의 재사용성이 향상된다.
- ② 자동화된 기법을 통해 소프트웨어 품질이 향상된다.
- ③ 소프트웨어 사용자들에게 사용 방법을 신속히 숙지시키기 위해 사용된다.
- ④ 소프트웨어 유지보수를 간편하게 수행할 수 있다.

[해설]

CASE는 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 도구로, 사용 방법의 신속한 숙지와는 무관합니다.

47. 익스트림 프로그래밍에 대한 설명으로 틀린 것은?

- ① 대표적인 구조적 방법론 중 하나이다.
- ② 소규모 개발 조직이 불확실하고 변경이 많은 요구를 접하였을 때 적절한 방법이다.
- ③ 익스트림 프로그래밍을 구동시키는 원리는 상식적인 원리와 경험을 최대한 끌어 올리는 것이다.
- ④ 구체적인 실천 방법을 정의하고 있으며, 개발 문서보다는 소스 코드에 중점을 둔다.

[해설]

익스트림 프로그래밍(XP; eXtreme Programming)은 애자일 개발 방법론을 기반으로 하는 소프트웨어 개발 모형입니다.

48. 객체지향 개념을 활용한 소프트웨어 구현과 관련한 설명 중 틀린 것은?

- ① 객체(Object)란 필요한 자료 구조와 수행되는 함수들을 가진 하나의 독립된 존재이다.
- ② JAVA에서 정보 은닉(Information Hiding)을 표기할 때 private의 의미는 '공개'이다.
- ③ 상속(Inheritance)은 개별 클래스를 상속 관계로 묶음으로써 클래스 간의 체계화된 전체 구조를 파악하기 쉽다는 장점이 있다.
- ④ 같은 클래스에 속하는 개개의 객체이자 하나의 클래스에서 생성된 객체를 인스턴스(Instance)라고 한다.

[해설]

JAVA에서 private는 외부로부터의 접근을 제한하는 접근 제어자로, 정보 은닉(Information Hiding)을 표기할 때 private의 의미는 '은닉'이고, '공개'를 의미하는 접근 제어자는 'Public'입니다.

49. 소프트웨어 공학에서 모델링(Modeling)과 관련한 설명으로 틀린 것은?

- ① 개발팀이 응용문제를 이해하는 데 도움을 줄 수 있다.
- ② 유지보수 단계에서만 모델링 기법을 활용한다.
- ③ 개발될 시스템에 대하여 여러 분야의 엔지니어들이 공통된 개념을 공유하는 데 도움을 준다.
- ④ 절차적인 프로그램을 위한 자료 흐름도는 프로세스 위주의 모델링 방법이다.

[해설]

모델링(Modeling)은 분석 및 설계 단계에서 개발하고자 하는 것을 시각적으로 표현한 것으로, 이렇게 제작된 모델은 소프트웨어 개발의 전 과정에서 지속적으로 사용됩니다.

50. 미들웨어(Middleware)에 대한 설명으로 틀린 것은?

- ① 여러 운영체제에서 응용 프로그램들 사이에 위치한 소프트웨어이다.
- ② 미들웨어의 서비스 이용을 위해 사용자가 정보 교환 방법 등의 내부 동작을 쉽게 확인할 수 있어야 한다.
- ③ 소프트웨어 컴포넌트를 연결하기 위한 준비된 인프라 구조를 제공한다.
- ④ 여러 컴포넌트를 1 대 1, 1 대 다, 다 대 다 등 여러 가지 형태로 연결이 가능하다.

[해설]

사용자가 미들웨어의 내부 동작을 확인하려면 별도의 응용 소프트웨어를 사용해야 하므로, 사용자가 미들웨어의 내부 동작을 확인하기는 쉽지 않습니다.

51. 모듈화(Modularity)와 관련한 설명으로 틀린 것은?

- ① 시스템을 모듈로 분할하면 각각의 모듈을 별개로 만들고 수정할 수 있기 때문에 좋은 구조가 된다.
- ② 응집도는 모듈과 모듈 사이의 상호 의존 또는 연관 정도를 의미한다.
- ③ 모듈 간의 결합도가 약해야 독립적인 모듈이 될 수 있다.
- ④ 모듈 내 구성 요소들 간의 응집도가 강해야 좋은 모듈 설계이다.

[해설]

- 응집도(Cohesion)는 명령어나 호출문 등 모듈의 내부 요소들의 서로 관련되어 있는 정도, 즉 모듈이 독립적인 기능으로 정의되어 있는 정도를 의미합니다.
- ②번은 결합도(Coupling)에 대한 설명입니다.

52. 요구사항 분석이 어려운 이유가 아닌 것은?

- ① 개발자와 사용자 간의 지식이나 표현의 차이가 커서 상호 이해가 쉽지 않다.
- ② 사용자의 요구는 예외가 거의 없어 열거와 구조화가 어렵지 않다.
- ③ 사용자의 요구사항이 모호하고 불명확하다.
- ④ 소프트웨어 개발 과정 중에 요구사항이 계속 변할 수 있다.

[해설]

요구사항(Requirement)은 예외가 많고, 모호하고 불명확하며, 중복 및 상충되는 사항들이 있을 뿐만 아니라, 개발 과정 중에도 수시로 변경될 수 있어 열거와 구조화가 어렵습니다.

53. 효과적인 모듈 설계를 위한 유의사항으로 거리가 먼 것은?

- ① 모듈간의 결합도를 약하게 하면 모듈 독립성이 향상된다.
- ② 복잡도와 중복성을 줄이고 일관성을 유지시킨다.
- ③ 모듈의 기능은 예측이 가능해야 하며 지나치게 제한적이어야 한다.
- ④ 유지보수가 용이해야 한다.

[해설]

모듈의 기능을 지나치게 제한적으로 설계할 필요는 없습니다.

54. HIPO(Hierarchy Input Process Output)에 대한 설명으로 거리가 먼 것은?

- ① **상향식 소프트웨어 개발을 위한 문서화 도구이다.**
- ② HIPO 차트 종류에는 가시적 도표, 총체적 도표, 세부적 도표가 있다.
- ③ 기능과 자료의 의존 관계를 동시에 표현할 수 있다.
- ④ 보기 쉽고 이해하기 쉽다.

[해설]

HIPO는 시스템의 분석 및 설계나 문서화할 때 사용되는 기법으로, 하향식 소프트웨어 개발을 위한 문서화 도구입니다.

55. 소프트웨어 개발 단계에서 요구 분석 과정에 대한 설명으로 거리가 먼 것은?

- ① 분석 결과의 문서화를 통해 향후 유지보수에 유용하게 활용할 수 있다.
- ② **개발 비용이 가장 많이 소요되는 단계이다.**
- ③ 자료 흐름도, 자료 사전 등이 효과적으로 이용될 수 있다.
- ④ 보다 구체적인 명세를 위해 소단위 명세서(Mini-Spec)가 활용될 수 있다.

[해설]

소프트웨어 생명 주기에서 개발 비용이 가장 많이 소요되는 단계는 유지보수 단계입니다.

56. 메시지 지향 미들웨어(MOM; Message-Oriented Middleware)에 대한 설명으로 틀린 것은?

- ① **느리고 안정적인 응답보다는 즉각적인 응답이 필요한 온라인 업무에 적합하다.**
- ② 독립적인 애플리케이션을 하나의 통합된 시스템으로 묶기 위한 역할을 한다.
- ③ 송신측과 수신측의 연결 시 메시지 큐를 활용하는 방법이 있다.
- ④ 상이한 애플리케이션 간 통신을 비동기 방식으로 지원한다.

[해설]

MOM은 온라인 업무보다는 이기종 분산 데이터 시스템의 데이터 동기를 위해 많이 사용됩니다.

57. 소프트웨어 설치 매뉴얼에 대한 설명으로 틀린 것은?

- ① 설치 과정에서 표시될 수 있는 예외상황에 관련 내용을 별도로 구분하여 설명한다.
- ② 설치 시작부터 완료할 때까지의 전 과정을 빠짐없이 순서대로 설명한다.
- ③ **설치 매뉴얼은 개발자 기준으로 작성한다.**
- ④ 설치 매뉴얼에는 목차, 개요, 기본사항 등이 기본적으로 포함되어야 한다.

[해설]

설치 매뉴얼을 보는 사람이 사용자이므로, 사용자 기준으로 작성해야 합니다.

58. 저작권 관리 구성 요소에 대한 설명이 틀린 것은?

- ① 콘텐츠 제공자(Contents Provider) : 콘텐츠를 제공하는 저작권자
- ② **콘텐츠 분배자(Contents Distributor) : 콘텐츠를 메타 데이터와 함께 배포 가능한 단위로 묶는 기능**
- ③ 클리어링 하우스(Clearing House) : 키 관리 및 라이선스 발급 관리
- ④ DRM 컨트롤러 : 배포된 콘텐츠의 이용 권한을 통제

[해설]

- 콘텐츠 분배자(Contents Distributor)는 암호화된 콘텐츠를 유통하는 곳이나 사람을 의미합니다.
- 콘텐츠를 메타 데이터와 함께 배포 가능한 단위로 묶는 기능을 수행하는 것은 패키지(Packager)입니다.

59. 단위 테스트(Unit Test)와 관련한 설명으로 틀린 것은?

- ① 구현 단계에서 각 모듈의 개발을 완료한 후 개발자가 명세서의 내용대로 정확히 구현되었는지 테스트 한다.
- ② 모듈 내부의 구조를 구체적으로 볼 수 있는 구조적 테스트를 주로 시행한다.
- ③ 필요 데이터를 인자를 통해 넘겨주고, 테스트 완료 후 그 결과값을 받는 역할을 하는 가상의 모듈을 테스트 스텝(Stub)이라고 한다.
- ④ 테스트할 모듈을 호출하는 모듈도 있고, 테스트할 모듈이 호출하는 모듈도 있다.

[해설]

- 테스트 스텝(Test Stub)은 제어 모듈이 호출하는 타 모듈의 기능을 단순히 수행하는 도구로, 일시적으로 필요한 조건만을 가지고 있는 시험용 모듈입니다.
- ③번은 테스트 드라이버(Test Driver)에 대한 설명입니다.

60. 소프트웨어 테스트와 관련한 설명으로 틀린 것은?

- ① 화이트박스 테스트는 모듈의 논리적인 구조를 체계적으로 점검할 수 있다.
- ② 블랙박스 테스트는 프로그램의 구조를 고려하지 않는다.
- ③ 테스트 케이스에는 일반적으로 시험 조건, 테스트 데이터, 예상 결과가 포함되어야 한다.
- ④ 화이트박스 테스트에서 기본 경로(Basis Path)란 흐름 그래프의 시작 노드에서 종료 노드까지의 서로 독립된 경로로 사이클을 허용하지 않는 경로를 말한다.

[해설]

기초 경로(Base Path = Basis Path)는 수행 가능한 모든 경로를 의미합니다.

61. 이진 검색 알고리즘에 대한 설명으로 틀린 것은?

- ① 탐색 효율이 좋고 탐색 시간이 적게 소요된다.
- ② 검색할 데이터가 정렬되어 있어야 한다.
- ③ 피보나치 수열에 따라 다음에 비교할 대상을 선정하여 검색한다.
- ④ 비교 횟수를 거듭할 때마다 검색 대상이 되는 데이터의 수가 절반으로 줄어든다.

[해설]

이진 검색과 피보나치 수열은 아무런 관계가 없습니다.

62. 소프트웨어 패키징 도구 활용 시 고려사항으로 틀린 것은?

- ① 반드시 내부 콘텐츠에 대한 암호화 및 보안을 고려한다.
- ② 보안을 위하여 이기종 연동을 고려하지 않아도 된다.
- ③ 사용자 편의성을 위한 복잡성 및 비효율성 문제를 고려한다.
- ④ 제품 소프트웨어 종류에 적합한 암호화 알고리즘을 적용한다.

[해설]

패키징 도구 활용 시에는 다른 여러 콘텐츠 및 단말기 간 연동을 고려해야 합니다.

63. 소프트웨어 패키징에 대한 설명으로 틀린 것은?

- ① 패키징은 개발자 중심으로 진행한다.
- ② 신규 및 변경 개발 소스를 식별하고, 이를 모듈화하여 상용제품으로 패키징 한다.
- ③ 고객의 편의성을 위해 매뉴얼 및 버전 관리를 지속적으로 한다.
- ④ 범용 환경에서 사용이 가능하도록 일반적인 배포 형태로 패키징이 진행된다.

[해설]

소프트웨어를 설계하거나 개발할 때 그리고 개발된 소프트웨어를 패키징 할 때까지도 모든 과정에서 가장 먼저 고려되어야 할 대상은 소프트웨어를 사용할 사용자입니다.

64. 연결 리스트(Linked List)에 대한 설명으로 거리가 먼 것은?

- ① 노드의 삽입이나 삭제가 쉽다.
- ② 노드들이 포인터로 연결되어 검색이 빠르다.
- ③ 연결을 해주는 포인터(Pointer)를 위한 추가 공간이 필요하다.
- ④ 연결 리스트 중에서 중간 노드 연결이 끊어지면 다음 노드를 찾기 힘들다.

[해설]

연결 리스트(Linked List)는 노드들이 포인터로 연결되어 포인터를 찾아가는 시간이 필요하므로 선형 리스트에 비해 검색 속도가 느립니다.

65. 클린 코드(Clean Code)를 작성하기 위한 원칙으로 틀린 것은?

- ① 추상화 : 하위 클래스/메소드/함수를 통해 애플리케이션의 특성을 간략하게 나타내고, 상세 내용은 상위 클래스/메소드/함수에서 구현한다.
- ② 의존성 : 다른 모듈에 미치는 영향을 최소화하도록 작성한다.
- ③ 가독성 : 누구든지 읽기 쉽게 코드를 작성한다.
- ④ 중복성 : 중복을 최소화할 수 있는 코드를 작성한다.

[해설]

추상화는 상위 클래스/메소드/함수를 통해 애플리케이션의 특성을 간략하게 나타내고, 상세 내용은 하위 클래스/메소드/함수에서 구현합니다.

66. 빌드 자동화 도구에 대한 설명으로 틀린 것은?

- ① Gradle은 실행할 처리 명령들을 모아 태스크로 만든 후 태스크 단위로 실행한다.
- ② 빌드 자동화 도구는 지속적인 통합 개발 환경에서 유용하게 활용된다.
- ③ 빌드 자동화 도구에는 Ant, Gradle, Jenkins 등이 있다.
- ④ Jenkins는 Groovy를 기반으로 한 오픈 소스로 안드로이드 앱 개발 환경에서 사용된다.

[해설]

- Jenkins는 Java 기반 오픈 소스 형태의 서버 서블릿 컨테이너에서 실행되는 서버 기반 도구입니다.
- ④번은 Gradle에 대한 설명입니다.

67. 테스트 드라이버(Test Driver)에 대한 설명으로 틀린 것은?

- ① 시험대상 모듈을 호출하는 간이 소프트웨어이다.
- ② 필요에 따라 매개 변수를 전달하고 모듈을 수행한 후의 결과를 보여줄 수 있다.
- ③ 상향식 통합 테스트에서 사용된다.
- ④ 테스트 대상 모듈이 호출하는 하위 모듈의 역할을 한다.

[해설]

비어있는 하위 모듈을 대체하는 것은 스텝(Stub), 상위 모듈을 대체하는 것은 드라이버(Driver)입니다.

68. White Box Testing에 대한 설명으로 옳지 않은 것은?

- ① Base Path Testing, Boundary Value Analysis가 대표적인 기법이다.
- ② Source Code의 모든 문장을 한 번 이상 수행함으로써 진행된다.
- ③ 모듈 안의 작동을 직접 관찰할 수 있다.
- ④ 산출물의 각 기능별로 적절한 프로그램의 제어 구조에 따라 선택, 반복 등의 부분들을 수행함으로써 논리적 경로를 점검한다.

[해설]

경계값 분석(Boundary Value Analysis)은 블랙박스 테스트의 종류에 해당합니다.

69. 테스트 케이스와 관련한 설명으로 틀린 것은?

- ① 테스트의 목표 및 테스트 방법을 결정하기 전에 테스트 케이스를 작성해야 한다.
- ② 프로그램에 결함이 있더라도 입력에 대해 정상적인 결과를 낼 수 있기 때문에 결함을 검사할 수 있는 테스트 케이스를 찾는 것이 중요하다.
- ③ 개발된 서비스가 정의된 요구사항을 준수하는지 확인하기 위한 입력 값과 실행 조건, 예상 결과의 집합으로 볼 수 있다.
- ④ 테스트 케이스 실행이 통과되었는지 실패하였는지 판단하기 위한 기준을 테스트 오라클(Test Oracle)이라고 한다.

[해설]

테스트 케이스는 테스트의 목표와 방법을 결정한 후 작성합니다.

70. 자료 구조에 대한 설명으로 틀린 것은?

- ① 큐는 비선형 구조에 해당한다.
- ② 큐는 First In-First Out 처리를 수행한다.
- ③ 스택은 Last In-First Out 처리를 수행한다.
- ④ 스택은 서브루틴 호출, 인터럽트 처리, 수식 계산 및 수식 표기법에 응용된다.

[해설]

큐(Queue)는 한쪽에서는 삽입이, 다른 한쪽에서는 인출이 이루어지는 선형 자료 구조입니다.

71. 소프트웨어 테스트에서 검증(Verification)과 확인(Validation)에 대한 설명으로 틀린 것은?

- ① 소프트웨어 테스트에서 검증과 확인을 구별하면 찾고자 하는 결함 유형을 명확하게 하는 데 도움이 된다.
- ② 검증은 소프트웨어 개발 과정을 테스트하는 것이고, 확인은 소프트웨어 결과를 테스트하는 것이다.
- ③ 검증은 작업 제품이 요구 명세의 기능, 비기능 요구사항을 얼마나 잘 준수하는지 측정하는 작업이다.
- ④ 검증은 작업 제품이 사용자의 요구에 적합한지 측정하며, 확인은 작업 제품이 개발자의 기대를 충족시키는지를 측정한다.

[해설]

- 검증(Verification)은 개발자의 입장에서 개발한 소프트웨어가 명세서에 맞게 만들어졌는지를 점검하는 것입니다.
- 확인(Validation)은 사용자의 입장에서 개발한 소프트웨어가 고객의 요구사항에 맞게 구현되었는지를 확인하는 것입니다.

72. DRM(Digital Rights Management)과 관련한 설명으로 틀린 것은?

- ① 디지털 콘텐츠와 디바이스의 사용을 제한하기 위해 하드웨어 제조업자, 저작권자, 출판업자 등이 사용할 수 있는 접근 제어 기술을 의미한다.
- ② 디지털 미디어의 생명 주기 동안 발생하는 사용 권한 관리, 과금, 유통 단계를 관리하는 기술로도 볼 수 있다.
- ③ 클리어링 하우스(Clearing House)는 사용자에게 콘텐츠 라이선스를 발급하고 권한을 부여해주는 시스템을 말한다.
- ④ 원본을 안전하게 유통하기 위한 전자적 보안은 고려하지 않기 때문에 불법 유통과 복제의 방지는 불가능하다.

[해설]

디지털 저작권 관리(DRM)는 콘텐츠를 안전하게 유통하기 위한 전자적 보안 장치인 보안 컨테이너(Security Container)를 통해 불법 유통과 복제를 방지할 수 있습니다.

73. 소프트웨어 형상 관리(Configuration Management)에 관한 설명으로 틀린 것은?

- ① 소프트웨어에서 일어나는 수정이나 변경을 알아내고 제어하는 것을 의미한다.
- ② 소프트웨어 개발의 전체 비용을 줄이고, 개발 과정의 여러 방해 요인이 최소화되도록 보증하는 것을 목적으로 한다.
- ③ 형상 관리를 위하여 구성된 팀을 “Chief Programmer Team”이라고 한다.
- ④ 형상 관리의 기능 중 하나는 버전 제어 기술이다.

[해설]

‘Chief Programmer Team’은 효율성을 증대시키기 위해 경험과 능력이 풍부한 책임 프로그래머를 중심으로 구성된 개발팀의 구성 방식 중 하나로 형상 관리와는 관계가 없습니다.

74. 화이트박스 테스트와 관련한 설명으로 틀린 것은?

- ① 화이트박스 테스트의 이해를 위해 논리 흐름도(Logic-Flow Diagram)를 이용할 수 있다.
- ② 테스트 데이터를 이용해 실제 프로그램을 실행함으로써 오류를 찾는 동적 테스트(Dynamic Test)에 해당한다.
- ③ 프로그램의 구조를 고려하지 않기 때문에 테스트 케이스는 프로그램 또는 모듈의 요구나 명세를 기초로 결정한다.
- ④ 테스트 데이터를 선택하기 위하여 검증 기준(Test Coverage)을 정한다.

[해설]

- 화이트박스 테스트는 프로그램의 제어 구조에 따라 선택, 반복 등의 분기점 부분들을 수행함으로써 논리적 경로를 제어합니다.
- ③번은 블랙박스 테스트에 대한 설명입니다.

75. IPSec(IP Security)에 대한 설명으로 틀린 것은?

- ① 암호화 수행시 일방향 암호화만 지원한다.
- ② ESP는 발신지 인증, 데이터 무결성, 기밀성 모두를 보장한다.
- ③ 운영 모드는 Tunnel 모드와 Transport 모드로 분류된다.
- ④ AH는 발신지 호스트를 인증하고, IP 패킷의 무결성을 보장한다.

[해설]

IPSec는 암호화와 복호화가 모두 가능한 양방향 암호 방식입니다.

76. 통합 테스트(Integration Test)와 관련한 설명으로 틀린 것은?

- ① 시스템을 구성하는 모듈의 인터페이스와 결합을 테스트하는 것이다.
- ② 하향식 통합 테스트의 경우 넓이 우선(Breadth First) 방식으로 테스트를 할 모듈을 선택할 수 있다.
- ③ 상향식 통합 테스트의 경우 시스템 구조도의 최상위에 있는 모듈을 먼저 구현하고 테스트한다.
- ④ 모듈 간의 인터페이스와 시스템의 동작이 정상적으로 잘되고 있는지를 빨리 파악하고자 할 때 상향식 보다는 하향식 통합 테스트를 사용하는 것이 좋다.

[해설]

상향식 통합 테스트는 프로그램의 하위 모듈에서 상위 모듈 방향으로 통합하면서 테스트하는 기법입니다.

77. 제품 소프트웨어의 형상 관리 역할로 틀린 것은?

- ① 형상 관리를 통해 이전 리버전이나 버전에 대한 정보에 접근 가능하여 배포본 관리에 유용
- ② 불필요한 사용자의 소스 수정 제한
- ③ 프로젝트 개발 비용을 효율적으로 관리
- ④ 동일한 프로젝트에 대해 여러 개발자 동시 개발 가능

[해설]

형상 관리(SCM)는 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동으로, 프로젝트 개발 비용의 효율적인 관리와는 무관합니다.

78. 스택에 대한 설명으로 틀린 것은?

- ① 입출력이 한쪽 끝으로만 제한된 리스트이다.
- ② Head(front)와 Tail(rear)의 2개 포인터를 갖고 있다.
- ③ LIFO 구조이다.
- ④ 더 이상 삭제할 데이터가 없는 상태에서 데이터를 삭제하면 언더플로(Underflow)가 발생한다.

[해설]

Front와 Rear의 2개의 포인터를 갖고 있는 자료 구조는 큐(Queue)입니다.

79. EAI(Enterprise Application Integration) 구축 유형 중 Hybrid에 대한 설명으로 틀린 것은?

- ① Hub & Spoke와 Message Bus의 혼합 방식이다.
- ② 필요한 경우 한 가지 방식으로 EAI 구현이 가능하다.
- ③ 데이터 병목 현상을 최소화할 수 있다.
- ④ 중간에 미들웨어를 두지 않고 각 애플리케이션을 Point to Point로 연결한다.

[해설]

Hybrid는 Hub & Spoke와 Message Bus의 혼합 방식으로, 단말들 사이에 버스와 허브를 미들웨어로 두어 애플리케이션을 연결합니다.

80. 다음 중 테스트 오라클에 대한 설명으로 옳지 않은 것은?

- ① 샘플링 오라클 : 특정한 몇몇 테스트 케이스의 입력 값들에 대해서만 기대하는 결과를 제공하는 오라클이다.
- ② 토탈 오라클 : 모든 테스트 케이스의 입력 값에 대해 기대하는 결과를 제공하는 오라클이다.
- ③ 휴리스틱 오라클 : 특정 테스트 케이스의 입력 값에 대해 기대하는 결과를 제공하고, 나머지 입력 값들에 대해서는 추정으로 처리하는 오라클이다.
- ④ 일관성 검사 오라클 : 애플리케이션의 변경이 있을 경우 테스트 케이스의 수행 전과 후의 결과 값이 동일한지를 확인하는 오라클이다.

[해설]

모든 테스트 케이스의 입력 값에 대해 기대하는 결과를 제공하는 오라클은 참 오라클(True Oracle)입니다.

81. 블랙박스 테스트를 이용하여 발견할 수 있는 오류가 아닌 것은?

- ① 비정상적인 자료를 입력해도 오류 처리를 수행하지 않는 경우
- ② 정상적인 자료를 입력해도 요구된 기능이 제대로 수행되지 않는 경우
- ③ 반복 조건을 만족하는데도 루프 내의 문장이 수행되지 않는 경우
- ④ 경계값을 입력할 경우 요구된 출력 결과가 나오지 않는 경우

[해설]

루프 내 문장의 수행 여부는 화이트박스 테스트를 통해서만 확인할 수 있습니다.

82. 코드의 간결성을 유지하기 위해 사용되는 지침으로 틀린 것은?

- ① 공백을 이용하여 실행문 그룹과 주석을 명확히 구분한다.
- ② 복잡한 논리식과 산술식은 괄호와 들여쓰기(Indentation)를 통해 명확히 표현한다.
- ③ 빈 줄을 사용하여 선언부와 구현부를 구별한다.
- ④ 한 줄에 최대한 많은 문장을 코딩한다.

[해설]

소스 코드는 가독성을 위해 줄 나눔과 들여쓰기, 괄호를 적절하게 사용해야 합니다.

83. 소프트웨어 형상 관리에 대한 설명으로 거리가 먼 것은?

- ① 소프트웨어에 가해지는 변경을 제어하고 관리한다.
- ② 프로젝트 계획, 분석서, 설계서, 프로그램, 테스트 케이스 모두 관리 대상이다.
- ③ 대표적인 형상 관리 도구로 Ant, Maven, Gradle 등이 있다.
- ④ 유지보수 단계뿐만 아니라 개발 단계에도 적용할 수 있다.

[해설]

- 형상 관리 도구에는 Git, CVS, Subversion, Mercurial 등이 있습니다.
- Ant, Maven, Gradle은 빌드 자동화 도구입니다.

84. IDE(Integrated Development Environment) 도구의 각 기능에 대한 설명으로 틀린 것은?

- ① Coding - 프로그래밍 언어를 가지고 컴퓨터 프로그램을 작성할 수 있는 환경을 제공
- ② Compile - 저급 언어의 프로그램을 고급 언어 프로그램으로 변환하는 기능
- ③ Debugging - 프로그램에서 발견되는 버그를 찾아 수정할 수 있는 기능
- ④ Deployment - 소프트웨어를 최종 사용자에게 전달하기 위한 기능

[해설]

컴파일(Compile)은 고급 언어로 작성한 프로그램을 컴퓨터가 이해할 수 있는 기계어(저급 언어)로 변환하는 기능입니다.

85. 클린 코드 작성원칙에 대한 설명으로 틀린 것은?

- ① 코드의 중복을 최소화한다.
- ② 코드가 다른 모듈에 미치는 영향을 최소화하도록 작성한다.
- ③ 누구든지 코드를 쉽게 읽을 수 있도록 작성한다.
- ④ 간단하게 코드를 작성한다.

[해설]

클린 코드(Clean Code)는 의존성 배제 원칙에 따라 코드가 다른 모듈에 미치는 영향을 최소화해야 합니다.

86. 알고리즘과 관련한 설명으로 틀린 것은?

- ① 주어진 작업을 수행하는 컴퓨터 명령어를 순서대로 나열한 것으로 볼 수 있다.
- ② 검색(Searching)은 정렬이 되지 않은 데이터 혹은 정렬이 된 데이터 중에서 키값에 해당되는 데이터를 찾는 알고리즘이다.
- ③ 정렬(Sorting)은 흩어져있는 데이터를 키값을 이용하여 순서대로 열거하는 알고리즘이다.
- ④ 선형 검색은 검색을 수행하기 전에 반드시 데이터의 집합이 정렬되어 있어야 한다.

[해설]

선형 검색(Linear Search)은 정렬이 되어 있지 않은 파일에서 순차적으로 검색하는 방식으로, 찾고자 하는 키(Key) 값을 첫 번째 레코드 키 값부터 차례로 비교하여 검색하는 방식입니다.

87. 소스 코드 정적 분석(Static Analysis)에 대한 설명으로 틀린 것은?

- ① 소스 코드를 실행시키지 않고 분석한다.
- ② 코드에 있는 오류나 잠재적인 오류를 찾아내기 위한 활동이다.
- ③ 하드웨어적인 방법으로만 코드 분석이 가능하다.
- ④ 자료 흐름이나 논리 흐름을 분석하여 비정상적인 패턴을 찾을 수 있다.

[해설]

소스 코드 정적 분석 도구 중 pmd, cppcheck는 소프트웨어적인 방법으로 코드를 분석합니다.

88. 힙 정렬(Heap Sort)에 대한 설명으로 틀린 것은?

- ① 정렬할 입력 레코드들로 힙을 구성하고 가장 큰 키 값을 갖는 루트 노드를 제거하는 과정을 반복하여 정렬하는 기법이다.
- ② 평균 수행 시간은 $O(n\log_2 n)$ 이다.
- ③ 완전 이진 트리(Complete Binary Tree)로 입력자료의 레코드를 구성한다.
- ④ 최악의 수행 시간은 $O(2n^4)$ 이다.

[해설]

힙 정렬의 평균과 최악 모두 시간 복잡도는 $O(n\log_2 n)$ 입니다.

89. 형상 관리의 개념과 절차에 대한 설명으로 틀린 것은?

- ① 형상 식별은 형상 관리 계획을 근거로 형상 관리의 대상이 무엇인지 식별하는 과정이다.
- ② 형상 관리를 통해 가시성과 추적성을 보장함으로써 소프트웨어의 생산성과 품질을 높일 수 있다.
- ③ **형상 통제 과정에서는 형상 목록의 변경 요구를 즉시 수용 및 반영해야 한다.**
- ④ 형상 감사는 형상 관리 계획대로 형상 관리가 진행되고 있는지, 형상 항목의 변경이 요구사항에 맞도록 제대로 이뤄졌는지 등을 살펴보는 활동이다.

[해설]

형상 통제 과정은 식별된 형상 항목에 대한 변경 요구를 검토하여 현재의 기준선(Base Line)이 잘 반영될 수 있도록 조정하는 작업입니다.

90. 정규화에 대한 설명으로 적절하지 않은 것은?

- ① **데이터베이스의 개념적 설계 단계 이전에 수행한다.**
- ② 데이터 구조의 안정성을 최대화한다.
- ③ 중복을 배제하여 삽입, 삭제, 갱신 이상의 발생을 방지한다.
- ④ 데이터 삽입 시 릴레이션을 재구성할 필요성을 줄인다.

[해설]

정규화는 개념적 설계 다음에 수행하는 논리적 설계 단계에서 수행하는 작업입니다.

91. 정규화의 필요성으로 거리가 먼 것은?

- ① 데이터 구조의 안정성 최대화
- ② **중복 데이터의 활성화**
- ③ 수정, 삭제 시 이상 현상의 최소화
- ④ 테이블 불일치 위험의 최소화

[해설]

정규화는 중복을 배제함으로써 삽입, 삭제, 갱신 이상의 발생을 최소화시킵니다.

92. 물리적 데이터베이스 설계에 대한 설명으로 거리가 먼 것은?

- ① 물리적 설계의 목적은 효율적인 방법으로 데이터를 저장하는 것이다.
- ② 트랜잭션 처리량과 응답시간, 디스크 용량 등을 고려해야 한다.
- ③ 저장 레코드의 형식, 순서, 접근 경로와 같은 정보를 사용하여 설계한다.
- ④ **트랜잭션의 인터페이스를 설계하며, 데이터 타입 및 데이터 타입들 간의 관계로 표현한다.**

[해설]

트랜잭션의 인터페이스를 설계하고 데이터 타입 및 타입들 간의 관계 표현은 논리 데이터베이스 설계에서 수행합니다.

93. 개체-관계(E-R) 모델에 대한 설명으로 잘못된 것은?

- ① 특정 DBMS를 고려하여 제작하지 않는다.
- ② **개체는 마름모, 속성은 사각형을 이용하여 표현한다.**
- ③ 개념적 데이터베이스 단계에서 제작된다.
- ④ E-R 모델의 기본적인 아이디어를 시각적으로 가장 잘 나타낸 것이 E-R 다이어그램이다.

[해설]

E-R 다이어그램에서 개체 타입은 사각형, 관계 타입은 마름모, 속성은 타원으로 표현합니다.

94. SQL 문에서 SELECT에 대한 설명으로 옳지 않은 것은?

- ① FROM 절에는 질의에 의해 검색될 데이터들을 포함하는 테이블명을 기술한다.
- ② 검색 결과에 중복되는 레코드를 없애기 위해서는 WHERE 절에 'DISTINCT' 키워드를 사용한다.
- ③ HAVING 절은 GROUP BY 절과 함께 사용되며, 그룹에 대한 조건을 지정한다.
- ④ ORDER BY 절은 특정 속성을 기준으로 정렬하여 검색할 때 사용한다.

[해설]

DISTINCT는 SELECT절의 속성명 앞에 사용하는 예약어입니다.

95. 뷰(View)에 대한 설명으로 옳지 않은 것은?

- ① 뷰는 CREATE 문을 사용하여 정의한다.
- ② 뷰는 데이터의 논리적 독립성을 제공한다.
- ③ 뷰를 제거할 때는 DROP 문을 사용한다.
- ④ 뷰는 저장장치 내에 물리적으로 존재한다.

[해설]

뷰(View)는 저장장치 내에 물리적으로 존재하지 않는 가상 테이블입니다.

96. 속성(attribute)에 대한 설명으로 틀린 것은?

- ① 속성은 개체의 특성을 기술한다.
- ② 속성은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- ③ 속성은 파일 구조상 데이터 항목 또는 데이터 필드에 해당된다.
- ④ 속성의 수를 "cardinality"라고 한다.

[해설]

속성(attribute)의 수는 디그리(Degree)이고, 카디널리티(Cardinality)는 튜플(Tuple)의 수입니다.

97. 뷰(VIEW)에 대한 설명으로 옳지 않은 것은?

- ① DBA는 보안 측면에서 뷰를 활용할 수 있다.
- ② 뷰 위에 또 다른 뷰를 정의할 수 있다.
- ③ 뷰에 대한 삽입, 갱신, 삭제 연산 시 제약사항이 따르지 않는다.
- ④ 독립적인 인덱스를 가질 수 없다.

[해설]

뷰는 기본 테이블이나 또 다른 뷰를 이용해서 만든 가상 테이블로서, 기본 테이블과 비교할 때 삽입, 삭제, 갱신 연산에 제약이 있습니다.

98. 데이터베이스의 무결성 규정(Integrity Rule)과 관련한 설명으로 틀린 것은?

- ① 무결성 규정에는 데이터가 만족해야 될 제약 조건, 규정을 참조할 때 사용하는 식별자 등의 요소가 포함될 수 있다.
- ② 무결성 규정의 대상으로는 도메인, 키, 종속성 등이 있다.
- ③ 정식으로 허가받은 사용자가 아닌 불법적인 사용자에게 의한 갱신으로부터 데이터베이스를 보호하기 위한 규정이다.
- ④ 릴레이션 무결성 규정(Relation Integrity Rules)은 릴레이션을 조작하는 과정에서의 의미적 관계(Semantic Relationship)를 명세한 것이다.

[해설]

허가받은 사용자만이 갱신할 수 있다는 설명은 데이터베이스의 무결성 규정(Integrity Rule)이 아닌 소프트웨어 개발 시 충족시켜야 할 보안 요소인 무결성(Integrity)에 대한 설명입니다.

99. 데이터베이스에서 병행제어의 목적으로 틀린 것은?

- ① 시스템 활용도 최대화
- ② 사용자에게 대한 응답시간 최소화
- ③ 데이터베이스 공유 최소화
- ④ 데이터베이스 일관성 유지

[해설]

병행제어의 목적 중 하나는 데이터베이스 공유 최대화입니다.

100. DELETE 명령에 대한 설명으로 틀린 것은?

- ① 테이블의 행을 삭제할 때 사용한다.
- ② WHERE 조건절이 없는 DELETE 명령을 수행하면 DROP TABLE 명령을 수행했을 때와 동일한 효과를 얻을 수 있다.
- ③ SQL을 사용 용도에 따라 분류할 경우 DML에 해당한다.
- ④ 기본 사용 형식은 “DELETE FROM 테이블 [WHERE 조건];” 이다.

[해설]

- DROP은 테이블을 삭제하고, DELETE는 레코드를 삭제하는 명령문입니다.
- DELETE에 WHERE 조건절을 생략하면 테이블은 남아있고 테이블 안에 있는 모든 레코드가 삭제됩니다.

101. 분산 데이터베이스 시스템(Distributed Database System)에 대한 설명으로 틀린 것은?

- ① 분산 데이터베이스는 논리적으로는 하나의 시스템에 속하지만, 물리적으로는 여러 개의 컴퓨터 사이트에 분산되어 있다.
- ② 위치 투명성, 중복 투명성, 병행 투명성, 장애 투명성을 목표로 한다.
- ③ 데이터베이스의 설계가 비교적 어렵고, 개발 비용과 처리 비용이 증가한다는 단점이 있다.
- ④ 분산 데이터베이스 시스템의 주요 구성 요소는 분산 처리기, P2P 시스템, 단일 데이터베이스 등이 있다.

[해설]

분산 데이터베이스 시스템의 주요 구성 요소는 분산 처리기, 분산 데이터베이스, 통신 네트워크가 있습니다.

102. 관계대수와 관계해석에 대한 설명으로 옳지 않은 것은?

- ① 관계대수는 원래 수학의 프레디킷 해석에 기반을 두고 있다.
- ② 관계대수로 표현한 식은 관계해석으로 표현할 수 있다.
- ③ 관계해석은 관계 데이터의 연산을 표현하는 방법이다.
- ④ 관계해석은 원하는 정보가 무엇이라는 것만 정의하는 비절차적인 특징을 가지고 있다.

[해설]

수학의 프레디킷 해석에 기반을 두고 있는 것은 관계해석입니다.

103. 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

- ① 시스템 자체에 관련 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스이다.
- ② 데이터 사전이라고도 한다.
- ③ 기본 테이블, 뷰, 인덱스, 패키지, 접근 권한 등의 정보를 저장한다.
- ④ 시스템을 위한 정보를 포함하는 시스템 데이터베이스이므로 일반 사용자는 SQL을 이용하여 내용을 검색해 볼 수 없다.

[해설]

카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 이용자로 SQL을 이용하여 내용을 검색해 볼 수 있습니다.

104. 데이터베이스의 인덱스와 관련한 설명으로 틀린 것은?

- ① 문헌의 색인, 사전과 같이 데이터를 쉽고 빠르게 찾을 수 있도록 만든 데이터 구조이다.
- ② 테이블에 붙여진 색인으로 데이터 검색 시 처리 속도 향상에 도움이 된다.
- ③ 인덱스의 추가, 삭제 명령어는 각각 ADD, DELETE이다.
- ④ 대부분의 데이터베이스에서 테이블을 삭제하면 인덱스도 같이 삭제된다.

[해설]

인덱스를 추가하는 명령어는 CREATE, 삭제하는 명령어는 DROP입니다.

105. 관계형 데이터 모델의 릴레이션에 대한 설명으로 틀린 것은?

- ① 모든 속성 값은 원자 값을 갖는다.
- ② 한 릴레이션에 포함된 튜플은 모두 상이하다.
- ③ 한 릴레이션에 포함된 튜플 사이에는 순서가 없다.
- ④ 한 릴레이션을 구성하는 속성 사이에는 순서가 존재한다.

[해설]

릴레이션 스키마를 구성하는 속성들 간의 순서는 중요하지 않으며, 특별한 순서가 없습니다.

106. 정규화를 거치지 않아 발생하게 되는 이상(anomaly) 현상의 종류에 대한 설명으로 옳지 않은 것은?

- ① 삭제 이상이란 릴레이션에서 한 튜플을 삭제할 때 의도와는 상관없는 값들로 함께 삭제되는 연쇄 삭제 현상이다.
- ② 삽입 이상이란 릴레이션에서 데이터를 삽입할 때 의도와는 상관없이 원하지 않는 값들로 함께 삽입되는 현상이다.
- ③ 갱신 이상이란 릴레이션에서 튜플에 있는 속성값을 갱신할 때 일부 튜플의 정보만 갱신되어 정보에 모순이 생기는 현상이다.
- ④ 종속 이상이란 하나의 릴레이션에 하나 이상의 함수적 종속성이 존재하는 현상이다.

[해설]

이상의 종류에는 삽입 이상, 삭제 이상, 갱신 이상이 있으며, 종속 이상은 존재하지 않습니다.

107. SQL의 TRUNCATE 명령어에 대한 설명으로 옳지 않은 것은?

- ① DELETE와 같이 테이블의 모든 데이터를 삭제한다.
- ② DROP과 달리 테이블 스키마는 제거되지 않고 유지된다.
- ③ DELETE에 비해 빠르게 데이터를 제거하는 것이 가능하다.
- ④ DELETE와 동일하게 ROLLBACK 명령어로 삭제된 데이터를 되살릴 수 있다.

[해설]

DELETE 명령어로 삭제한 데이터는 ROLLBACK 명령어로 되살릴 수 있지만, TRUNCATE 명령어로 삭제한 데이터는 되살릴 수 없습니다.

108. 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

- ① 사용자가 직접 시스템 카탈로그의 내용을 갱신하여 데이터베이스 무결성을 유지한다.
- ② 시스템 자신이 필요로 하는 스키마 및 여러 가지 객체에 관한 정보를 포함하고 있는 시스템 데이터베이스이다.
- ③ 시스템 카탈로그에 저장되는 내용을 메타 데이터라고도 한다.
- ④ 시스템 카탈로그는 DBMS가 스스로 생성하고 유지한다.

[해설]

시스템 카탈로그는 사용자가 조회할 수는 있으나 갱신하는 것은 불가능합니다.

109. 데이터베이스에서 개념적 설계 단계에 대한 설명으로 틀린 것은?

- ① 산출물로 E-R Diagram을 만들 수 있다.
- ② DBMS에 독립적인 개념 스키마를 설계한다.
- ③ 트랜잭션 인터페이스를 설계 및 작성한다.
- ④ 논리적 설계 단계의 앞 단계에서 수행된다.

[해설]

트랜잭션의 인터페이스 설계는 논리적 설계 단계에서 수행하는 작업입니다.

110. 관계대수에 대한 설명으로 틀린 것은?

- ① 원하는 릴레이션을 정의하는 방법을 제공하며 비절차적 언어이다.
- ② 릴레이션 조작을 위한 연산의 집합으로 피연산자와 결과가 모두 릴레이션이다.
- ③ 일반 집합 연산과 순수 관계 연산으로 구분된다.
- ④ 질의에 대한 해를 구하기 위해 수행해야 할 연산의 순서를 명시한다.

[해설]

원하는 릴레이션을 정의하는 방법을 제공하는 비절차적 특성을 지닌 것은 관계해석입니다.

111. 데이터베이스의 논리적 설계(Logical Design) 단계에서 수행하는 작업이 아닌 것은?

- ① 레코드 집종의 분석 및 설계
- ② 논리적 데이터베이스 구조로 매핑(mapping)
- ③ 트랜잭션 인터페이스 설계
- ④ 스키마의 평가 및 정제

[해설]

레코드 집종의 분석은 물리적 설계 단계에서 수행하는 작업입니다.

112. 데이터 제어 언어(DCL)의 기능으로 옳지 않은 것은?

- ① 데이터 보안
- ② 논리적, 물리적 데이터 구조 정의
- ③ 무결성 유지
- ④ 병행수행 제어

[해설]

논리적, 물리적 데이터 구조를 정의하는 것은 데이터 정의 언어(DDL; Data Definition Language)의 기능입니다.

113. CREATE TABLE문에 포함되지 않는 기능은?

- ① 속성 타입 변경
- ② 속성의 NOT NULL 여부 지정
- ③ 기본키를 구성하는 속성 지정
- ④ CHECK 제약조건의 정의

[해설]

CREATE TABLE문에서는 속성 타입을 변경할 수 없습니다.

114. 무결성을 보장하기 위해 트랜잭션이 가져야 할 특성에 대한 설명으로 옳지 않은 것은?

- ① 트랜잭션 내의 모든 명령은 반드시 완벽히 수행되어야 하며, 모두가 완벽히 수행되지 않고 어느 하나라도 오류가 발생하면 트랜잭션 전부가 취소되어야 한다.
- ② 트랜잭션의 수행과 관계 없이 데이터베이스가 가지고 있는 고정요소는 일관되어야 한다.
- ③ 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션의 연산이 끼어들 수 없다.
- ④ Commit과 Rollback 명령어에 의해 보장받는 트랜잭션의 특성은 일관성이다.

[해설]

Commit과 Rollback 명령어에 의해 보장받는 트랜잭션의 특성은 일관성이 아니라 원자성입니다.

115. 데이터베이스 설계 시 물리적 설계 단계에서 수행하는 사항이 아닌 것은?

- ① 저장 레코드 양식 설계
- ② 레코드 집종의 분석 및 설계
- ③ 접근 경로 설계
- ④ 목표 DBMS에 맞는 스키마 설계

[해설]

목표 DBMS에 맞는 스키마 설계는 논리적 설계 단계에서 수행하는 사항입니다.

116. 분산 데이터베이스 시스템과 관련한 설명으로 틀린 것은?

- ① 물리적으로 분산된 데이터베이스 시스템을 논리적으로 하나의 데이터베이스 시스템처럼 사용할 수 있도록 한 것이다.
- ② 물리적으로 분산되어 지역별로 필요한 데이터를 처리할 수 있는 지역 컴퓨터(Local Computer)를 분산 처리기(Distributed Processor)라고 한다.
- ③ 분산 데이터베이스 시스템을 위한 통신 네트워크 구조가 데이터 통신에 영향을 주므로 효율적으로 설계해야 한다.
- ④ 데이터베이스가 분산되어 있음을 사용자가 인식할 수 있도록 분산 투명성(Distribution Transparency)을 배제해야 한다.

[해설]

사용자가 데이터베이스가 분산되어 있음을 인식할 필요가 없습니다.

117. 테이블의 기본키(Primary Key)로 지정된 속성에 관한 설명으로 가장 거리가 먼 것은?

- ① NOT NULL로 널 값을 가지지 않는다.
- ② 릴레이션에서 튜플을 구별할 수 있다.
- ③ 외래키로 참조될 수 있다.
- ④ 검색할 때 반드시 필요하다.

[해설]

기본키가 지정되어 있지 않아도 검색할 수 있습니다.

118. SQL과 관련한 설명으로 틀린 것은?

- ① REVOKE 키워드를 사용하여 열 이름을 다시 부여할 수 있다.
- ② 데이터 정의어는 기본 테이블, 뷰 테이블, 또는 인덱스 등을 생성, 변경, 제거하는데 사용되는 명령어이다.
- ③ DISTINCT를 활용하여 중복 값을 제거할 수 있다.
- ④ JOIN을 통해 여러 테이블의 레코드를 조합하여 표현할 수 있다.

[해설]

REVOKE는 데이터베이스 사용자의 사용 권한을 취소하는 기능입니다.

119. 데이터베이스에서 인덱스(Index)와 관련한 설명으로 틀린 것은?

- ① 인덱스의 기본 목적은 검색 성능을 최적화하는 것으로 볼 수 있다.
- ② B-트리 인덱스는 분기를 목적으로 하는 Branch Block을 가지고 있다.
- ③ BETWEEN 등 범위(Range) 검색에 활용될 수 있다.
- ④ 시스템이 자동으로 생성하여 사용자가 변경할 수 없다.

[해설]

인덱스는 사용자가 데이터 정의어(DDL)를 이용하여 생성, 변경, 제거할 수 있습니다.

120. 병행제어 기법 중 로킹에 대한 설명으로 옳지 않은 것은?

- ① 로킹의 대상이 되는 객체의 크기를 로킹 단위라고 한다.
- ② 데이터베이스, 파일, 레코드 등은 로킹 단위가 될 수 있다.
- ③ 로킹의 단위가 작아지면 로킹 오버헤드가 증가한다.
- ④ 로킹의 단위가 커지면 데이터베이스 공유도가 증가한다.

[해설]

로킹 단위가 커지면 데이터베이스 병행성 수준이 낮아지므로 공유도가 감소합니다.

121. C언어에서의 변수 선언으로 틀린 것은?

- ① `int else;`
- ② `int Test2;`
- ③ `int pc;`
- ④ `int True;`

[해설]

`else`는 if문에서 사용하는 예약어로, C언어에서는 예약어를 변수의 이름으로 사용할 수 없습니다.

122. 페이징 기법에서 페이지 크기가 작아질수록 발생하는 현상이 아닌 것은?

- ① 기억장소 이용 효율이 증가한다.
- ② 입·출력 시간이 늘어난다.
- ③ 내부 단편화가 감소한다.
- ④ 페이지 맵 테이블의 크기가 감소한다.

[해설]

페이지 크기가 작아질수록 페이지의 개수가 많아져 주소를 저장하는 맵 테이블의 크기가 커지게 됩니다.

123. RIP 라우팅 프로토콜에 대한 설명으로 틀린 것은?

- ① 경로 선택 메트릭은 홉 카운트(hop count)이다.
- ② 라우팅 프로토콜을 IGP와 EGP로 분류했을 때 EGP에 해당한다.
- ③ 최단 경로 탐색에 Bellman-Ford 알고리즘을 사용한다.
- ④ 각 라우터는 이웃 라우터들로부터 수신한 정보를 이용하여 라우팅 표를 갱신한다.

[해설]

RIP 라우팅 프로토콜은 IGP(내부 게이트웨이 프로토콜)에 해당합니다.

124. IP 주소 체계와 관련한 설명으로 틀린 것은?

- ① IPv6의 패킷 헤더는 32 octet의 고정된 길이를 가진다.
- ② IPv6는 주소 자동설정(Auto Configuration) 기능을 통해 손쉽게 이용자의 단말을 네트워크에 접속시킬 수 있다.
- ③ IPv4는 호스트 주소를 자동으로 설정하며 유니캐스트(Unicast)를 지원한다.
- ④ IPv4는 클래스별로 네트워크와 호스트 주소의 길이가 다르다.

[해설]

IPv6의 패킷 헤더는 40옥텟(octet)의 고정된 길이를 갖습니다.

125. UNIX 운영체제에 관한 특징으로 틀린 것은?

- ① 하나 이상의 작업에 대하여 백그라운드에서 수행이 가능하다.
- ② Multi-User는 지원하지만 Multi-Tasking은 지원하지 않는다.
- ③ 트리 구조의 파일 시스템을 갖는다.
- ④ 이식성이 높으며 장치 간의 호환성이 높다.

[해설]

UNIX는 다중 사용자(Multi-User), 다중 작업(Multi-Tasking)을 지원하는 운영체제입니다.

126. JAVA에서 변수와 자료형에 대한 설명으로 틀린 것은?

- ① 변수는 어떤 값을 주기억장치에 기억하기 위해서 사용하는 공간이다.
- ② 변수의 자료형에 따라 저장할 수 있는 값의 종류와 범위가 달라진다.
- ③ char 자료형은 나열된 여러 개의 문자를 저장하고자 할 때 사용한다.
- ④ boolean 자료형은 조건이 참인지 거짓인지 판단하고자 할 때 사용한다.

[해설]

char 자료형은 문자 한 글자를 저장할 때 사용하는 자료형입니다.

127. UNIX 시스템의 셸(Shell)의 주요 기능에 대한 설명이 아닌 것은?

- ① 사용자 명령을 해석하고 커널로 전달하는 기능을 제공한다.
- ② 반복적인 명령 프로그램을 만드는 프로그래밍 기능을 제공한다.
- ③ 셸 프로그램 실행을 위해 프로세스와 메모리를 관리한다.
- ④ 초기화 파일을 이용해 사용자 환경을 설정하는 기능을 제공한다.

[해설]

프로세스와 메모리를 관리하는 것은 커널(Kernel)의 기능입니다.

128. 프로세스 적재 정책과 관련한 설명으로 틀린 것은?

- ① 반복, 스택, 부프로그램은 시간 지역성(Temporal Locality)과 관련이 있다.
- ② 공간 지역성(Spatial Locality)은 프로세스가 어떤 페이지를 참조했다면 이후 가상주소 공간상 그 페이지와 인접한 페이지들을 참조할 가능성이 높음을 의미한다.
- ③ 일반적으로 페이지 교환에 보내는 시간보다 프로세스 수행에 보내는 시간이 더 크면 스레싱(Thrashing)이 발생한다.
- ④ 스레싱(Thrashing) 현상을 방지하기 위해서는 각 프로세스가 필요로 하는 프레임을 제공할 수 있어야 한다.

[해설]

스레싱(Thrashing)은 프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상입니다.

129. TCP 헤더와 관련한 설명으로 틀린 것은?

- ① 순서 번호(Sequence Number)는 전달하는 바이트마다 번호가 부여된다.
- ② 수신 번호 확인(Acknowledgement Number)은 상대방 호스트에서 받으려는 바이트의 번호를 정의한다.
- ③ 체크섬(Checksum)은 데이터를 포함한 세그먼트의 오류를 검사한다.
- ④ 윈도우 크기는 송수신 측의 버퍼 크기로 최대 크기는 32767bit이다.

[해설]

TCP 헤더에서 윈도우의 최대 크기는 $65,535(2^{16}-1)$ byte입니다.

130. JAVA의 예외(exception)와 관련한 설명으로 틀린 것은?

- ① 문법 오류로 인해 발생한 것
- ② 오동작이나 결과에 악영향을 미칠 수 있는 실행 시간 동안에 발생한 오류
- ③ 배열의 인덱스가 그 범위를 넘어서는 경우 발생하는 오류
- ④ 존재하지 않는 파일을 읽으려고 하는 경우에 발생하는 오류

[해설]

- 예외(Exception)는 실행 중에 발생할 수 있는 여러 상황들을 대비한 것입니다.
- 문법 오류의 경우 코드가 실행조차 되지 않으므로 예외로 처리할 수 없습니다.

131. TCP 프로토콜에 대한 설명으로 거리가 먼 것은?

- ① 신뢰성이 있는 연결 지향형 전달 서비스이다.
- ② 기본 헤더 크기는 100byte이고 160byte까지 확장 가능하다.
- ③ 스트림 전송 기능을 제공한다.
- ④ 순서 제어, 오류 제어, 흐름 제어 기능을 제공한다

[해설]

TCP 프로토콜 헤더의 크기는 20~60byte이며, 여기에 선택적으로 40byte를 더 추가할 수 있으므로 최대 100byte까지 확장이 가능합니다.

132. IPv6에 대한 설명으로 틀린 것은?

- ① 32비트의 주소 체계를 사용한다.
- ② 멀티미디어의 실시간 처리가 가능하다.
- ③ IPv4보다 보안성이 강화되었다.
- ④ 자동으로 네트워크 환경 구성이 가능하다.

[해설]

IPv6의 주소 체계는 16비트씩 8부분, 총 128비트로 구성되어 있습니다.

133. 파이썬의 변수 작성 규칙 설명으로 옳지 않은 것은?

- ① 첫 자리에 숫자를 사용할 수 없다.
- ② 영문 대문자/소문자, 숫자, 밑줄(_)의 사용이 가능하다.
- ③ 변수 이름의 중간에 공백을 사용할 수 있다.
- ④ 이미 사용되고 있는 예약어는 사용할 수 없다.

[해설]

변수 이름의 중간에 공백을 넣을 수 없습니다.

134. C언어에서 사용할 수 없는 변수명은?

- ① student2019
- ② text-color
- ③ _korea
- ④ amount

[해설]

변수명에는 공백이나 *, +, -, / 등의 특수문자를 사용할 수 없습니다.

135. 다음 중 HRN에 대한 설명으로 옳지 않은 것은?

- ① 대기 시간과 서비스 시간을 이용하는 방법이다.
- ② 대기 시간이 긴 프로세스일 경우 우선순위가 높다.
- ③ 우선순위 계산식 값이 낮을수록 우선순위가 높다.
- ④ SJF 기법을 보완하기 위한 스케줄링 방법이다.

[해설]

HRN 기법의 우선순위 계산식을 통해 산출된 값은 수치가 클수록 우선순위가 높습니다.

136. 프로세스와 관련한 설명으로 틀린 것은?

- ① 프로세스가 준비 상태에서 프로세서가 배당되어 실행 상태로 변화하는 것을 디스패치(Dispatch)라고 한다.
- ② 프로세스 제어 블록(PCB, Process Control Block)은 프로세스 식별자, 프로세스 상태 등의 정보로 구성된다.
- ③ 이전 프로세스의 상태 레지스터 내용을 보관하고 다른 프로세스의 레지스터를 적재하는 과정을 문맥 교환(Context Switching)이라고 한다.
- ④ 프로세스는 스레드(Thread) 내에서 실행되는 흐름의 단위이며, 스레드와 달리 주소 공간에 실행 스택(Stack)이 없다.

[해설]

스레드(Thread)가 프로세스 내에서의 작업의 단위입니다.

137. 다음 중 프로세스에 대한 설명 중 틀린 것은?

- ① 프로세서가 할당되는 실체로, 디스패치가 가능한 단위이다.
- ② 프로세스는 비동기적 행위를 일으키는 주체이다.
- ③ 프로세스는 스레드 내의 작업 단위를 의미하며, 경량 스레드라고도 불린다.
- ④ PCB를 가지며 PCB에는 프로세스의 현재상태, 고유식별자를 가지고 있다.

[해설]

- ③번은 프로세스와 스레드를 반대로 설명하고 있습니다.
- 스레드는 프로세스 내의 작업 단위를 의미하며, 경량 프로세스라고도 불립니다.

138. IPv6에 대한 특성으로 틀린 것은?

- ① 표시 방법은 8비트씩 4부분의 10진수로 표시한다.
- ② 2^{128} 개의 주소를 표현할 수 있다.
- ③ 등급별, 서비스별로 패킷을 구분할 수 있어 품질 보장이 용이하다.
- ④ 확장 기능을 통해 보안 기능을 제공한다.

[해설]

IPv6는 16비트씩 8부분, 총 128비트로 구성되어 있으며 각 부분을 16진수로 표현합니다.

139. 파일 디스크립터(File Descriptor)에 대한 설명으로 틀린 것은?

- ① 파일 관리를 위해 시스템이 필요로 하는 정보를 가지고 있다.
- ② 보조기억장치에 저장되어 있다가 파일이 개방(open)되면 주기억장치로 이동된다.
- ③ 사용자가 파일 디스크립터를 직접 참조할 수 있다.
- ④ 파일 제어 블록(File Control Block)이라고도 한다.

[해설]

파일 디스크립터는 사용자가 직접 참조할 수 없습니다.

140. UDP 프로토콜의 특징이 아닌 것은?

- ① 비연결형 서비스를 제공한다.
- ② 단순한 헤더 구조로 오버헤드가 적다.
- ③ 주로 주소를 지정하고, 경로를 설정하는 기능을 한다.
- ④ TCP와 같이 트랜스포트 계층에 존재한다.

[해설]

③번은 IP(Internet Protocol)에 대한 설명입니다.

141. UNIX의 셸(Shell)에 관한 설명으로 옳지 않은 것은?

- ① 명령어 해석기이다.
- ② 시스템과 사용자 간의 인터페이스를 담당한다.
- ③ 여러 종류의 셸이 있다.
- ④ 프로세스, 기억장치, 입출력 관리를 수행한다.

[해설]

프로세스 관리, 기억장치 관리, 파일 관리 등의 작업을 수행하는 것은 커널(Kernel)입니다.

142. 라이브러리의 개념과 구성에 대한 설명 중 틀린 것은?

- ① 라이브러리란 필요할 때 찾아서 쓸 수 있도록 모듈화되어 제공되는 프로그램을 말한다.
- ② 프로그래밍 언어에 따라 일반적으로 도움말, 설치 파일, 샘플 코드 등을 제공한다.
- ③ 외부 라이브러리는 프로그래밍 언어가 기본적으로 가지고 있는 라이브러리를 의미하며, 표준 라이브러리는 별도의 파일 설치를 필요로 하는 라이브러리를 의미한다.
- ④ 라이브러리는 모듈과 패키지를 총칭하며, 모듈이 개별 파일이라면 패키지는 파일들을 모아 놓은 폴더라고 볼 수 있다.

[해설]

프로그래밍 언어가 기본적으로 가지고 있는 라이브러리가 표준 라이브러리, 별도의 파일 설치를 필요로 하는 라이브러리는 외부 라이브러리입니다.

143. C언어에서 malloc() 함수에 대한 설명으로 틀린 것은?

- ① 원하는 시점에 원하는 만큼 메모리를 동적으로 할당한다.
- ② 사용자가 입력한 bit만큼 메모리를 할당한다.
- ③ free 명령어로 할당된 메모리를 해제한다.
- ④ 메모리 할당이 불가능할 경우 NULL이 반환된다.

[해설]

malloc() 함수는 입력한 Byte만큼 메모리를 할당하는 함수입니다.

144. IP 프로토콜의 주요 특징에 해당하지 않는 것은?

- ① 체크섬(Checksum) 기능으로 데이터 체크섬(Data Checksum)만 제공한다.
- ② 패킷을 분할, 병합하는 기능을 수행하기도 한다.
- ③ 비연결형 서비스를 제공한다.
- ④ Best Effort 원칙에 따른 전송 기능을 제공한다.

[해설]

IP(Internet Protocol)는 헤더 체크섬(Header Checksum)만 제공합니다.

145. OSI 7계층 중 네트워크 계층에 대한 설명으로 틀린 것은?

- ① 패킷을 발신지로부터 최종 목적지까지 전달하는 책임을 진다.
- ② 한 노드로부터 다른 노드로 프레임을 전송하는 책임을 진다.
- ③ 패킷에 발신지와 목적지의 논리 주소를 추가한다.
- ④ 라우터 또는 교환기는 패킷 전달을 위해 경로를 지정하거나 교환 기능을 제공한다.

[해설]

- 네트워크 계층의 프로토콜 데이터 단위(PDU)는 패킷(Packet)입니다.
- PDU가 프레임(Frame)인 계층은 데이터 링크 계층입니다.

146. 배치 프로그램의 필수 요소에 대한 설명으로 틀린 것은?

- ① 자동화는 심각한 오류 상황 외에는 사용자의 개입 없이 동작해야 한다.
- ② 안정성은 어떤 문제가 생겼는지, 언제 발생했는지 등을 추적할 수 있어야 한다.
- ③ 대용량 데이터는 대용량의 데이터를 처리할 수 있어야 한다.
- ④ 무결성은 주어진 시간 내에 처리를 완료할 수 있어야 하고, 동시에 동작하고 있는 다른 애플리케이션을 방해하지 말아야 한다.

[해설]

- ④번은 배치 프로그램의 필수 요소 중 '성능'에 대한 설명입니다.

147. TCP 프로토콜과 관련한 설명으로 틀린 것은?

- ① 인접한 노드 사이의 프레임 전송 및 오류를 제어한다.
- ② 흐름 제어(Flow Control)의 기능을 수행한다.
- ③ 전이중(Full Duplex) 방식의 양방향 가상 회선을 제공한다.
- ④ 전송 데이터와 응답 데이터를 함께 전송할 수 있다.

[해설]

- TCP는 패킷의 전송 및 오류를 제어합니다.
- 프레임의 전송 및 오류 제어는 데이터 링크 계층의 프로토콜인 HDLC, LAPB, LLC, MAC 등이 수행합니다.

148. IPv6에 대한 설명으로 틀린 것은?

- ① 멀티캐스팅(Multicast) 대신 브로드캐스트(Broadcast)를 사용한다.
- ② 보안과 인증 확장 헤더를 사용함으로써 인터넷 계층의 보안 기능을 강화하였다.
- ③ 애니캐스트(Anycast)는 하나의 호스트에서 그룹 내의 가장 가까운 곳에 있는 수신자에게 전달하는 방식이다.
- ④ 128비트 주소 체계를 사용한다.

[해설]

IPv6는 유니캐스트, 멀티캐스트, 애니캐스트의 3가지 방식의 주소 체계를 사용합니다.

149. 리눅스 Bash 셸(Shell)에서 export와 관련한 설명으로 틀린 것은?

- ① 변수를 출력하고자 할 때는 export를 사용해야 한다.
- ② export가 매개 변수 없이 쓰일 경우 현재 설정된 환경 변수들이 출력된다.
- ③ 사용자가 생성하는 변수는 export 명령어로 표시하지 않는 한 현재 셸에 국한된다.
- ④ 변수를 export 시키면 전역(Global) 변수처럼 되어 끝까지 기억된다.

[해설]

export는 기존 환경 변수의 값을 변경하거나, 새로운 환경 변수를 설정할 때 사용하는 명령어입니다.

150. 운영체제에 대한 설명으로 거리가 먼 것은?

- ① 다중 사용자와 다중 응용 프로그램 환경 하에서 자원의 현재 상태를 파악하고 자원 분배를 위한 스케줄링을 담당한다.
- ② CPU, 메모리 공간, 기억 장치, 입출력 장치 등의 자원을 관리한다.
- ③ 운영체제의 종류로는 매크로 프로세서, 어셈블러, 컴파일러 등이 있다.
- ④ 입·출력장치와 사용자 프로그램을 제어한다.

[해설]

운영체제의 종류에는 Windows, Unix/Linux, MacOS, Android 등이 있습니다.

151. 다음 중 IP 버전에 대한 설명 중 틀린 것은?

- ① IPv4는 각 부분을 옥텟으로 구성, 총 32비트로 구성된다.
- ② IPv6는 각 부분을 콜론으로 구분한다.
- ③ IPv4는 네트워크 부분의 길이에 따라 A 클래스에서 E 클래스까지 총 5단계로 구성되어 있다.
- ④ IPv6는 IPv4에 비해 자료 전송 속도가 느리다.

[해설]

IPv6는 IPv4에 비해 자료 전송 속도가 빠릅니다.

152. 소프트웨어 개발 프레임워크를 적용할 경우 기대 효과로 거리가 먼 것은?

- ① 품질 보증
- ② 시스템 복잡도 증가
- ③ 개발 용이성
- ④ 변경 용이성

[해설]

소프트웨어 개발 프레임워크를 적용할 경우 시스템 복잡도의 증가가 아닌 감소 효과를 기대할 수 있습니다.

153. Windows 파일 시스템인 FAT와 비교했을 때의 NTFS의 특징이 아닌 것은?

- ① 보안에 취약
- ② 대용량 볼륨에 효율적
- ③ 자동 압축 및 안정성
- ④ 저용량 볼륨에서의 속도 저하

[해설]

NTFS는 FAT 또는 FAT32에 비해 성능, 보안, 안정성 면에서 뛰어납니다.

154. 소프트웨어 개발 프레임워크와 관련한 설명으로 가장 적절하지 않은 것은?

- ① 반제품 상태의 제품을 토대로 도메인별로 필요한 서비스 컴포넌트를 사용하여 재사용성 확대와 성능을 보장받을 수 있게 하는 개발 소프트웨어이다.
- ② 라이브러리와는 달리 사용자 코드에서 프레임워크를 호출해서 사용하고, 그에 대한 제어도 사용자 코드가 가지는 방식이다.
- ③ 설계 관점에 개발 방식을 패턴화시키기 위한 노력의 결과물인 소프트웨어 디자인 패턴을 반제품 소프트웨어 상태로 집적화시킨 것으로 볼 수 있다.
- ④ 프레임워크의 동작 원리를 그 제어 흐름의 일반적인 프로그램 흐름과 반대로 동작한다고 해서 IoC(Inversion of Control)이라고 설명하기도 한다.

[해설]

사용자 코드가 제어를 갖고 직접 호출해서 사용하는 것은 라이브러리(Library)입니다.

155. 해쉬(Hash) 기법에 대한 설명으로 틀린 것은?

- ① 임의의 길이의 입력 데이터를 받아 고정된 길이의 해쉬 값으로 변환한다.
- ② 주로 공개키 암호화 방식에서 키 생성을 위해 사용한다.
- ③ 대표적인 해쉬 알고리즘으로 HAVAL, SHA-1 등이 있다.
- ④ 해쉬 함수는 일방향 함수(One-way function)이다.

[해설]

공개키 암호화 알고리즘은 해시 기법이 아닌 소인수 분해나 이산대수 등 고유의 방법으로 키를 생성합니다.

156. Secure OS의 보안 기능으로 거리가 먼 것은?

- ① 식별 및 인증
- ② 임의적 접근 통제
- ③ 고가용성 지원
- ④ 강제적 접근 통제

[해설]

Secure OS의 보안 기능에는 식별 및 인증, 임의적/강제적 접근 통제, 객체 재사용 보호, 완전한 조정, 신뢰 경로, 감사 및 감사기록 축소 등이 있습니다.

157. 소프트웨어 개발 방법론의 테일러링(Tailoring)과 관련한 설명으로 틀린 것은?

- ① 프로젝트 수행 시 예상되는 변화를 배제하고 신속히 진행하여야 한다.
- ② 프로젝트에 최적화된 개발 방법론을 적용하기 위해 절차, 산출물 등을 적절히 변경하는 활동이다.
- ③ 관리 측면에서의 목적 중 하나는 최단기간에 안정적인 프로젝트 진행을 위한 사전 위험을 식별하고 제거하는 것이다.
- ④ 기술적 측면에서의 목적 중 하나는 프로젝트에 최적화된 기술 요소를 도입하여 프로젝트 특성에 맞는 최적의 기법과 도구를 사용하는 것이다.

[해설]

테일러링(Tailoring)은 프로젝트 상황 및 특성에 맞추어 기존의 방법론, 프로세스 등을 수정하는 것이니만큼, 예상되는 변화를 충분히 고려해야 합니다.

158. SPICE 모델의 프로세스 수행 능력 수준의 단계별 설명이 틀린 것은?

- ① 수준 7 - 미완성 단계
- ② 수준 5 - 최적화 단계
- ③ 수준 4 - 예측 단계
- ④ 수준 3 - 확립 단계

[해설]

SPICE 모델의 프로세스 수행 능력 수준은 '불완전(0), 수행(1), 관리(2), 확립(3), 예측(4), 최적화(5)'의 6단계로 구분합니다.

159. 공개키 암호에 대한 설명으로 틀린 것은?

- ① 10명이 공개키 암호를 사용할 경우 5개의 키가 필요하다.
- ② 복호화키는 비공개 되어 있다.
- ③ 송신자는 수신자의 공개키로 문서를 암호화한다.
- ④ 공개키 암호로 널리 알려진 알고리즘은 RSA가 있다.

[해설]

공개키 암호화 방식에서 키의 개수는 $2n$ 개로, 10명이 사용할 경우 20개의 키가 필요합니다.

160. COCOMO 모델에 의한 비용 산정에 대한 설명으로 옳지 않은 것은?

- ① 보헴이 제안한 원시 프로그램의 규모에 의한 비용 예측 모형이다.
- ② 같은 규모의 소프트웨어라도 그 유형에 따라 비용이 다르게 산정된다.
- ③ 비용 산정 유형으로 Organic Mode, Embedded Mode, Semi-Detached Mode가 있다.
- ④ UFP(Unadjusted Function Point)를 계산한다.

[해설]

UFP(Unadjusted Function Point)는 기능 점수(Function Point) 모델에서 기능 점수를 산출하는 과정에서 계산되는 값입니다.

161. 소프트웨어 공학에 대한 설명으로 거리가 먼 것은?

- ① 소프트웨어 공학이란 소프트웨어의 개발, 운용, 유지보수 및 파기에 대한 체계적인 접근 방법이다.
- ② 소프트웨어 공학은 소프트웨어 제품의 품질을 향상시키고 소프트웨어 생산성과 작업 만족도를 증대시키는 것이 목적이다.
- ③ 소프트웨어 공학의 궁극적 목표는 최대의 비용으로 계획된 일정보다 가능한 빠른 시일 내에 소프트웨어를 개발하는 것이다.
- ④ 소프트웨어 공학은 신뢰성 있는 소프트웨어를 경제적인 비용으로 획득하기 위해 공학적 원리를 정립하고 이를 이용하는 것이다.

[해설]

소프트웨어 공학은 소프트웨어의 품질과 생산성 향상을 목적으로 합니다.

162. 정보 보안을 위한 접근 제어(Access Control)와 관련한 설명으로 틀린 것은?

- ① 적절한 권한을 가진 인가자만 특정 시스템이나 정보에 접근할 수 있도록 통제하는 것이다.
- ② 시스템 및 네트워크에 대한 접근 제어의 가장 기본적인 수단은 IP와 서비스 포트로 볼 수 있다.
- ③ DBMS에 보안 정책을 적용하는 도구인 XDMCP를 통해 데이터베이스에 대한 접근 제어를 수행할 수 있다.
- ④ 네트워크 장비에서 수행하는 IP에 대한 접근 제어로는 관리 인터페이스의 접근 제어와 ACL(Access Control List) 등이 있다.

[해설]

XDMCP(X Display Manager Control Protocol)는 GUI 환경을 구축한 UNIX나 LINUX 시스템에서 서버와 클라이언트가 GUI 환경 관리자인 XDM(X Display Manager)과 네트워크를 통해 소통하는데 사용하는 프로토콜입니다.

163. 대칭 암호 알고리즘과 비대칭 암호 알고리즘에 대한 설명으로 틀린 것은?

- ① 대칭 암호 알고리즘은 비교적 실행 속도가 빠르기 때문에 다양한 암호의 핵심 함수로 사용될 수 있다.
- ② 대칭 암호 알고리즘은 비밀키 전달을 위한 키 교환이 필요하지 않아 암호화 및 복호화의 속도가 빠르다.
- ③ 비대칭 암호 알고리즘은 자신만이 보관하는 비밀키를 이용하여 인증, 전자서명 등에 적용이 가능하다.
- ④ 대표적인 대칭키 암호 알고리즘으로는 AES, IDEA 등이 있다.

[해설]

대칭 암호 알고리즘은 비밀키(Private Key)를 공유해야 하기 때문에 키 교환이 필요합니다.

164. IPSec(IP Security)에 대한 설명으로 틀린 것은?

- ① 암호화 수행 시 일방향 암호화만 지원한다.
- ② ESP는 발신지 인증, 데이터 무결성, 기밀성 모두를 보장한다.
- ③ 운영 모드는 Tunnel 모드와 Transport 모드로 분류된다.
- ④ AH는 발신지 호스트를 인증하고, IP 패킷의 무결성을 보장한다.

[해설]

IPSec는 암호화와 복호화가 모두 가능한 양방향 암호 방식입니다.

165. CBD(Component Based Development)에 대한 설명으로 틀린 것은?

- ① 개발 기간 단축으로 인한 생산성 향상
- ② 새로운 기능 추가가 쉬운 확장성
- ③ 소프트웨어 재사용이 가능
- ④ 1960년대까지 가장 많이 적용되었던 소프트웨어 개발 방법

[해설]

④번은 구조적 방법론에 대한 설명입니다.

166. 소프트웨어 개발 방법론 중 CBD(Component Based Development)에 대한 설명으로 틀린 것은?

- ① 생산성과 품질을 높이고, 유지보수 비용을 최소화할 수 있다.
- ② 컴포넌트 제작 기법을 통해 재사용성을 향상시킨다.
- ③ 모듈의 분할과 정복에 의한 하향식 설계방식이다.
- ④ 독립적인 컴포넌트 단위의 관리로 복잡성을 최소화할 수 있다.

[해설]

분할과 정복(Divide and Conquer) 원리를 사용하는 설계 방법론은 구조적 방법론입니다.

167. NS(Nassi-Schneiderman) Chart에 대한 설명으로 거리가 먼 것은?

- ① 논리의 기술에 중점을 둔 도형식 표현 방법이다.
- ② 연속, 선택 및 다중 선택, 반복 등의 제어논리 구조로 표현한다.
- ③ 주로 화살표를 사용하여 논리적인 제어 구조로 흐름을 표현한다.
- ④ 조건이 복합되어 있는 곳의 처리를 시각적으로 명확히 식별하는데 적합하다.

[해설]

화살표를 사용하여 논리적인 제어 구조로 흐름을 표현하는 표기법은 흐름도(Flowchart)입니다.

168. 공개키 암호화 방식에 대한 설명으로 틀린 것은?

- ① 공개키로 암호화된 메시지는 반드시 공개키로 복호화 해야 한다.
- ② 비대칭 암호기법이라고도 한다.
- ③ 대표적인 기법은 RSA 기법이 있다.
- ④ 키 분배가 용이하고, 관리해야 할 키 개수가 적다.

[해설]

공개키 암호화 기법은 암호화 할 때는 공개키(Public Key)를, 복호화 할 때는 비밀키(Secret Key)를 사용합니다.

169. 소프트웨어 정의 데이터센터(SDDC: Software Defined Data Center)에 대한 설명으로 틀린 것은?

- ① 컴퓨팅, 네트워킹, 스토리지, 관리 등을 모두 소프트웨어로 정의한다.
- ② 인력 개입 없이 소프트웨어 조작만으로 자동 제어 관리한다.
- ③ 데이터센터 내 모든 자원을 가상화하여 서비스한다.
- ④ 특정 하드웨어에 종속되어 특화된 업무를 서비스하기에 적합하다.

[해설]

소프트웨어 정의 데이터센터(SDDC)는 데이터센터의 모든 자원을 가상화하여 서비스하므로 특정 하드웨어와 상관없이 독립적으로 서비스를 제공할 수 있습니다.

170. 클라우드 기반 HSM(Cloud-based Hardware Security Module)에 대한 설명으로 틀린 것은?

- ① 클라우드(데이터센터) 기반 암호화 키 생성, 처리, 저장 등을 하는 보안 기기이다.
- ② 국내에서는 공인인증제의 폐지와 전자서명법 개정을 추진하면서 클라우드 HSM 용어가 자주 등장하였다.
- ③ 클라우드에 인증서를 저장하므로 기존 HSM 기기나 휴대폰에 인증서를 저장해 다닐 필요가 없다.
- ④ 하드웨어가 아닌 소프트웨어적으로만 구현되기 때문에 소프트웨어식 암호 기술에 내재된 보안 취약점을 해결할 수 없다는 것이 주요 단점이다.

[해설]

클라우드 기반 HSM은 암호화 키 생성이 하드웨어적으로 구현되므로 소프트웨어적으로 구현된 암호 기술이 가지는 보안 취약점을 무시할 수 있습니다.

171. 라우팅 프로토콜인 OSPF(Open Shortest Path First)에 대한 설명으로 옳지 않은 것은?

- ① 네트워크 변화에 신속하게 대처할 수 있다.
- ② **거리 벡터 라우팅 프로토콜이라고 한다.**
- ③ 멀티캐스팅을 지원한다.
- ④ 최단 경로 탐색에 Dijkstra 알고리즘을 사용한다.

[해설]

거리 벡터 라우팅 프로토콜이라고 불리는 것은 RIP(Routing Information Protocol)입니다.

172. 스트림 암호화 방식의 설명으로 옳지 않은 것은?

- ① 비트/바이트/단어들을 순차적으로 암호화한다.
- ② **해쉬 함수를 이용한 해쉬 암호화 방식을 사용한다.**
- ③ RC4는 스트림 암호화 방식에 해당한다.
- ④ 대칭키 암호화 방식이다.

[해설]

해쉬 암호화 방식은 블록 단위로 해쉬 알고리즘을 적용하는 방식으로, 스트림 암호화 방식과는 무관합니다.

173. 취약점 관리를 위한 응용 프로그램의 보안 설정과 가장 거리가 먼 것은?

- ① **서버 관리실 출입 통제**
- ② 실행 프로세스 권한 설정
- ③ 운영체제의 접근 제한
- ④ 운영체제의 정보 수집 제한

[해설]

서버 관리실 출입 통제는 물리적 보안에 포함되는 보안 조치입니다.

174. 소프트웨어 생명 주기 모형 중 Spiral Model에 대한 설명으로 틀린 것은?

- ① 비교적 대규모 시스템에 적합하다.
- ② 개발 순서는 계획 및 정의, 위험 분석, 공학적 개발, 고객 평가 순으로 진행된다.
- ③ 소프트웨어를 개발하면서 발생할 수 있는 위험을 관리하고 최소화하는 것을 목적으로 한다.
- ④ **계획, 설계, 개발, 평가의 개발 주기가 한 번만 수행된다.**

[해설]

나선형 모델(Spiral Model)은 계획 수립, 위험 분석, 개발 및 검증, 고객 평가 과정을 반복하며 수행하는 개발 방법론입니다.

175. SSH(Secure Shell)에 대한 설명으로 틀린 것은?

- ① **SSH의 기본 네트워크 포트는 220번을 사용한다.**
- ② 전송되는 데이터는 암호화 된다.
- ③ 키를 통한 인증은 클라이언트의 공개키를 서버에 등록해야 한다.
- ④ 서로 연결되어 있는 컴퓨터 간 원격 명령 실행이나 셸 서비스 등을 수행한다.

[해설]

SSH(Secure Shell)의 기본 네트워크 포트는 22번입니다.

176. CPM(Critical Path Method)에 대한 설명으로 옳지 않은 것은?

- ① 프로젝트 내에서 각 작업이 수행되는 시간과 각 작업 사이의 관계를 파악할 수 있다.
- ② **작업 일정을 한눈에 볼 수 있도록 해주며 막대 그래프의 형태로 표현한다.**
- ③ 효과적인 프로젝트의 통제를 가능하게 해 준다.
- ④ 경영층의 과학적인 의사 결정을 지원한다.

[해설]

작업 일정을 한눈에 볼 수 있도록 해주며 막대 그래프의 형태로 표현하는 것은 간트 차트(Gantt Chart)입니다.

177. 간트 차트(Gantt Chart)에 대한 설명으로 틀린 것은?

- ① 프로젝트를 이루는 소작업 별로 언제 시작되고 언제 끝나야 하는지를 한 눈에 볼 수 있도록 도와준다.
- ② 자원 배치 계획에 유용하게 사용된다.
- ③ CPM 네트워크로부터 만드는 것이 가능하다.
- ④ **수평 막대의 길이는 각 작업(Task)에 필요한 인원수를 나타낸다.**

[해설]

간트 차트(Gantt Chart)에서 수평 막대의 길이는 각 작업(Task)의 기간을 나타냅니다.

178. 소프트웨어 생명 주기 모델 중 V 모델과 관련한 설명으로 틀린 것은?

- ① **요구 분석 및 설계 단계를 거치지 않으며 항상 통합 테스트를 중심으로 V 형태를 이룬다.**
- ② Perry에 의해 제안되었으며 세부적인 테스트 과정으로 구성되어 신뢰도 높은 시스템을 개발하는 데 효과적이다.
- ③ 개발 작업과 검증 작업 사이의 관계를 명확히 들어내 놓은 폭포수 모델의 변형이라고 볼 수 있다.
- ④ 폭포수 모델이 산출물 중심이라면 V 모델은 작업과 결과의 검증에 초점을 둔다.

[해설]

소프트웨어 생명 주기의 V-모델은 '요구사항 → 분석 → 설계 → 구현' 단계로 수행되며 각 단계를 테스트와 연결하여 표현합니다.

179. 폭포수 모형의 특징으로 거리가 먼 것은?

- ① **개발 중 발생한 요구사항을 쉽게 반영할 수 있다.**
- ② 순차적인 접근 방법을 이용한다.
- ③ 단계적 정의와 산출물이 명확하다.
- ④ 모형의 적용 경험과 성공사례가 많다.

[해설]

폭포수 모형은 이전 단계로 돌아갈 수 없으므로 개발 중 발생한 요구사항을 쉽게 반영할 수 없습니다.

180. 소프트웨어 개발 프레임워크의 적용 효과로 볼 수 없는 것은?

- ① 공통 컴포넌트 재사용으로 중복 예산 절감
- ② **기술 종속으로 인한 선행 사업자 의존도 증대**
- ③ 표준화된 연계 모듈 활용으로 상호 운용성 향상
- ④ 개발표준에 의한 모듈화로 유지보수 용이

[해설]

프레임워크를 통한 표준화된 개발 기반은 선행 사업자의 의존도를 감소시키며, 사업자 종속성을 해소합니다.

181. RIP(Routing Information Protocol)에 대한 설명으로 틀린 것은?

- ① 거리 벡터 라우팅 프로토콜이라고도 한다.
- ② 소규모 네트워크 환경에 적합하다.
- ③ **최대 홉 카운트를 115홉 이하로 한정하고 있다.**
- ④ 최단경로탐색에는 Bellman-Ford 알고리즘을 사용한다.

[해설]

RIP의 최대 홉 수는 115가 아닌 15입니다.

182. SQL Injection 공격과 관련한 설명으로 틀린 것은?

- ① SQL Injection은 임의로 작성한 SQL 구문을 애플리케이션에 삽입하는 공격 방식이다.
- ② SQL Injection 취약점이 발생하는 곳은 주로 웹 애플리케이션과 데이터베이스가 연동되는 부분이다.
- ③ **DBMS의 종류와 관계없이 SQL Injection 공격 기법은 모두 동일하다.**
- ④ 로그인과 같이 웹에서 사용자의 입력값을 받아 데이터베이스 SQL문으로 데이터를 요청하는 경우 SQL Injection을 수행할 수 있다.

[해설]

SQL 삽입(SQL Injection) 공격은 웹 응용 프로그램에 SQL을 삽입하여 내부 데이터베이스(DB) 서버의 데이터를 유출 및 변조하고 관리자 인증을 우회하는 기법으로, DBMS의 종류에 따라 접근하는 방법이 달라지므로 공격 기법이 모두 동일하다는 말은 잘못되었습니다.

183. 물리적 위협으로 인한 문제에 해당하지 않는 것은?

- ① 화재, 홍수 등 천재지변으로 인한 위협
- ② 하드웨어 파손, 고장으로 인한 장애
- ③ 방화, 테러로 인한 하드웨어와 기록장치를 물리적으로 파괴하는 행위
- ④ **방화벽 설정의 잘못된 조작으로 인한 네트워크, 서버 보안 위협**

[해설]

방화벽 설정의 잘못된 조작으로 인한 네트워크, 서버 보안 위협은 기술적 위협으로 인한 문제에 해당합니다.

184. 침입 탐지 시스템(IDS; Intrusion Detection System)과 관련한 설명으로 틀린 것은?

- ① **이상 탐지 기법(Anomaly Detection)은 Signature Base나 Knowledge Base라고도 불리며 이미 발견되고 정립된 공격 패턴을 입력해두었다가 탐지 및 차단한다.**
- ② HIDS(Host-Based Intrusion Detection)는 운영체제에 설정된 사용자 계정에 따라 어떤 사용자가 어떤 접근을 시도하고 어떤 작업을 했는지에 대한 기록을 남기고 추적한다.
- ③ NIDS(Network-Based Intrusion Detection System)로는 대표적으로 Snort가 있다.
- ④ 외부 인터넷에 서비스를 제공하는 서버가 위치하는 네트워크인 DMZ(Demilitarized Zone)에는 IDS가 설치될 수 있다.

[해설]

- 이상 탐지 기법(Anomaly Detection)은 평균적인 시스템의 상태를 기준으로 비정상적인 행위나 자원의 사용이 감지되면 이를 알려주는 시스템입니다.
- ①번은 오용 탐지 기법(Misuse Detection)에 대한 설명입니다.

※ 이후 문제는 옳은 문장을 찾는 문제입니다.

185. C Class에 속하는 IP address는?

- ① 200.168.30.1
- ② 10.3.2.14
- ③ 225.2.4.1
- ④ 172.16.98.3

[해설]

C Class에 속하는 IP address의 범위는 192.0.0.0 ~ 223.255.255.255까지입니다.

186. 시스템에서 모듈 사이의 결합도(Coupling)에 대한 설명으로 옳은 것은?

- ① 한 모듈 내에 있는 처리 요소들 사이의 기능적인 연관 정도를 나타낸다.
- ② 결합도가 높으면 시스템 구현 및 유지보수 작업이 쉽다.
- ③ 모듈 간의 결합도를 약하게 하면 모듈 독립성이 향상된다.
- ④ 자료 결합도는 내용 결합도보다 결합도가 높다.

[해설]

- ① 결합도는 모듈 상호 간에 의존하는 정도를 의미합니다. 한 모듈 내에 있는 처리 요소들 사이의 기능적인 연관 정도를 나타내는 것은 응집도입니다.
- ② 결합도가 높으면 시스템 구현 및 유지보수 작업이 어렵습니다.
- ④ 자료 결합도는 내용 결합도보다 결합도가 낮습니다.

187. 스레드(Thread)에 대한 설명으로 옳지 않은 것은?

- ① 한 개의 프로세스는 여러 개의 스레드를 가질 수 없다.
- ② 커널 스레드의 경우 운영체제에 의해 스레드를 운용한다.
- ③ 사용자 스레드의 경우 사용자가 만든 라이브러리를 사용하여 스레드를 운용한다.
- ④ 스레드를 사용함으로써 하드웨어, 운영체제의 성능과 응용 프로그램의 처리율을 향상시킬 수 있다.

[해설]

한 개의 프로세스는 여러 개의 스레드를 가질 수 있습니다.

188. 프레임워크(Framework)에 대한 설명으로 옳은 것은?

- ① 소프트웨어 구성에 필요한 기본 구조를 제공함으로써 재사용이 가능하게 해준다.
- ② 소프트웨어 개발 시 구조가 잡혀 있기 때문에 확장이 불가능하다.
- ③ 소프트웨어 아키텍처(Architecture)와 동일한 개념이다.
- ④ 모듈화(Modularity)가 불가능하다.

[해설]

- ②, ④ 프레임워크는 모듈화가 잘 되어 있어 자유로운 확장이 가능합니다.
- ③ 프레임워크는 소프트웨어 아키텍처와 다릅니다.

189. 스택(Stack)에 대한 옳은 내용으로만 나열된 것은?

- ㉠ FIFO 방식으로 처리된다.
- ㉡ 순서 리스트의 뒤(Rear)에서 노드가 삽입되며, 앞(Front)에서 노드가 제거된다.
- ㉢ 선형 리스트의 양쪽 끝에서 삽입과 삭제가 모두 가능한 자료 구조이다.
- ㉣ 인터럽트 처리, 서브루틴 호출 작업 등에 응용된다.

- ① ㉠, ㉡
- ② ㉡, ㉢
- ③ ㉢
- ④ ㉠, ㉡, ㉢, ㉣

[해설]

- ㉠ 스택은 후입선출(LIFO; Last In First Out) 방식으로 자료를 처리합니다.
- ㉡은 큐(Queue), ㉢은 데크(Deque)에 대한 설명입니다.

190. 사용자 수준에서 지원되는 스레드(thread)가 커널에서 지원되는 스레드에 비해 가지는 장점으로 옳은 것은?

- ① 한 프로세스가 운영체제를 호출할 때 전체 프로세스가 대기할 필요가 없으므로 시스템 성능을 높일 수 있다.
- ② 동시에 여러 스레드가 커널에 접근할 수 있으므로 여러 스레드가 시스템 호출을 동시에 사용할 수 있다.
- ③ 각 스레드를 개별적으로 관리할 수 있으므로 스레드의 독립적인 스케줄링이 가능하다.
- ④ 커널 모드로의 전환 없이 스레드 교환이 가능하므로 오버헤드가 줄어든다.

[해설]

- ①, ②, ③번은 커널 수준 스레드에 대한 설명입니다.

191. 유스케이스(Usecase)에 대한 설명 중 옳은 것은?

- ① 유스케이스 다이어그램은 개발자의 요구를 추출하고 분석하기 위해 주로 사용한다.
- ② 액터는 대상 시스템과 상호 작용하는 사람이나 다른 시스템에 의한 역할이다.
- ③ 사용자 액터는 본 시스템과 데이터를 주고받는 연동 시스템을 의미한다.
- ④ 연동의 개념은 일방적으로 데이터를 파일이나 정해진 형식으로 넘겨주는 것을 의미한다.

[해설]

- ① 유스케이스 다이어그램은 사용자의 요구를 분석하는 데 사용합니다.
- ③ 사용자 액터(주액터)는 시스템을 사용함으로써 이득을 얻는 대상을 의미합니다. 본 시스템과 데이터를 주고받는 연동 시스템은 시스템 액터(부액터)입니다.
- ④ 연동은 2개 이상의 시스템이 일방적이 아닌 상호 간의 동작에 영향을 줄 수 있도록 연결망을 구성하는 것을 의미합니다.

192. DES는 몇 비트의 암호화 알고리즘인가?

- ① 8
- ② 24
- ③ 64
- ④ 132

[해설]

- 암호화 알고리즘이 몇 비트냐고 묻는 것은 한 번에 암호화하는 블록의 크기를 묻는 것입니다.
- DES(Data Encryption Standard)의 블록 크기는 64비트입니다.

193. 무결성 제약 조건 중 개체 무결성 제약 조건에 대한 설명으로 옳은 것은?

- ① 릴레이션 내의 튜플들이 각 속성의 도메인에 정해진 값만을 가져야 한다.
- ② 기본키는 NULL 값을 가져서는 안되며 릴레이션 내에 오직 하나의 값만 존재해야 한다.
- ③ 자식 릴레이션의 외래키는 부모 릴레이션의 기본키와 도메인이 동일해야 한다.
- ④ 자식 릴레이션의 값이 변경될 때 부모 릴레이션의 제약을 받는다.

[해설]

- 개체 무결성 제약 조건은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정입니다.
- ①번은 도메인 무결성, ③, ④번은 참조 무결성에 대한 설명입니다.

194. 프로젝트 일정 관리 시 사용하는 PERT 차트에 대한 설명에 해당하는 것은?

- ① 각 작업들이 언제 시작하고 언제 종료되는지에 대한 일정을 막대 도표를 이용하여 표시한다.
- ② 시간선(Time-Line) 차트라고도 한다.
- ③ 수평 막대의 길이는 각 작업의 기간을 나타낸다.
- ④ 작업들 간의 상호 관련성, 결정경로, 경계시간, 자원할당 등을 제시한다.

[해설]

- ①, ②, ③번은 간트 차트(Gantt Chart)에 대한 설명입니다.

195. 요구사항 분석에서 비기능적(Nonfunctional) 요구에 대한 설명으로 옳은 것은?

- ① 시스템의 처리량(Throughput), 반응 시간 등의 성능 요구나 품질 요구는 비기능적 요구에 해당하지 않는다.
- ② '차량 대여 시스템이 제공하는 모든 화면이 3초 이내에 사용자에게 보여야 한다'는 비기능적 요구이다.
- ③ 시스템 구축과 관련된 안전, 보안에 대한 요구사항들은 비기능적 요구에 해당하지 않는다.
- ④ '금융 시스템은 조회, 인출, 입금, 송금의 기능이 있어야 한다'는 비기능적 요구이다.

[해설]

- ① 성능 요구나 품질 요구는 비기능 요구사항에 해당합니다.
- ②번은 비기능 요구사항 중 성능 요구사항에 해당합니다.
- ③ 안전이나 보안에 대한 요구사항은 비기능 요구사항에 해당합니다.
- ④번은 기능 요구사항에 해당합니다.

196. UDP 특성에 해당되는 것은?

- ① 양방향 연결형 서비스를 제공한다.
- ② 송신중에 링크를 유지 관리하므로 신뢰성이 높다.
- ③ 순서 제어, 오류 제어, 흐름 제어 기능을 한다.
- ④ 흐름 제어나 순서 제어가 없어 전송 속도가 빠르다.

[해설]

- ① UDP는 비연결형 서비스를 제공합니다.
- ②, ③ UDP는 흐름 제어나 순서 제어가 없어 전송 속도가 빨라 신뢰성보다는 속도가 중요시되는 네트워크에서 사용됩니다.

197. 관계 데이터 모델에서 릴레이션(relation)에 관한 설명으로 옳은 것은?

- ① 릴레이션의 각 행을 스키마(Schema)라 하며, 예로 도서 릴레이션을 구성하는 스키마에는 도서번호, 도서명, 저자, 가격 등이 있다.
- ② 릴레이션의 각 열을 튜플(Tuple)이라 하며, 하나의 튜플은 각 속성에서 정의된 값을 이용하여 구성된다.
- ③ 도메인(Domain)은 하나의 속성이 가질 수 있는 같은 타입의 모든 값의 집합으로 각 속성의 도메인은 원자값을 갖는다.
- ④ 속성(Attribute)은 한 개의 릴레이션의 논리적인 구조를 정의한 것으로 릴레이션의 이름과 릴레이션에 포함된 속성들의 집합을 의미한다.

[해설]

- ① 릴레이션의 각 행은 튜플이라고 불리며, 스키마는 데이터베이스의 구조와 제약 조건에 대한 명세를 의미합니다.
- ② 릴레이션의 각 열을 속성이라고 합니다.
- ④ 릴레이션의 논리적인 구조를 정의한 것으로 릴레이션에 포함된 속성들의 집합을 릴레이션 스키마라고 합니다.